Università degli Studi di Padova
Facoltà di Ingegneria

Corso di Laurea Magistrale in Ingegneria Elettronica

Tesi di Laurea Magistrale

A second order continuous time $\Delta\Sigma$ modulator for ultra low power applications

Candidato:
Matteo Pagin
Matricola 1056922

Relatore:
Andrea Bevilacqua

Anno Accademico 2014–2015

Contents

List of Figures

Abstract

In this work the analysis and design of a continuous time 2nd order $\Delta\Sigma$ modulator for ultra low power applications is presented. The demand for low power application is raising and especially the RF part of a circuit is one of the most power hungry having an ADC that won't require big amounts of power to run is then required. Continuous time $\Delta\Sigma$ have become more popular in the last decade for this very reason, they are a suitable choice for low power applications and their continuous time architecture has an inherent anti aliasing filter which can completely remove the need for a dedicated filter. In this work a loop filter using only one amplifier is used, and the compensation of the amplifier is made such not to waste current in charging extra capacitance. Also a successive approximation register is used as the internal quantizer hence making the whole architecture designed for low power applications. The modulator achieve a maximum SNR of $65dB$ with an input bandwidth of $500kHZ$ and consuming only $69\mu W$ per conversion. The circuit was simulated using a $65nm$ CMOS process and will be fabricated later this year.

Ringraziamenti

Per la realizzazione di questa tesi vorrei ringraziare Pietro Andreani che ha supervisionato il lavoro e mi ha ospitato nel suo dipartimento presso l'università di Lund (Svezia); la sua guida è stata di grande aiuto durante il lavoro. Grazie anche al Prof. Andrea Bevilacqua che mi ha permesso di scrivere la tesi all'estero e mi ha seguito nella stesura di questa tesi. Grazie inoltre a tutti quelli che mi hanno supportato (e sopportato molte volte) e accompagnato durante questi anni di studio, in particolare a mamma e papà che mi han permesso di seguire un percorso di studi universitario.

Padova, Aprile 2015 Matteo Pagin

Introduction

In this work I will present the study and designing of an analog to digital converter(ADC) to be used for ultra low power applications. As the electronic circuits keep permeating everyday life and their performance increasing at high speed rate power consumption is becoming always more and more a key aspect of designing electronic circuits. In areas like smart sensors and biomedical implants the power consumption of the device is critical, as those are often isolated systems and replacing them, or their batteries, might be hard or impossible thus we strive to achieve a lifetime of several years for these applications. In order to do so it's necessary to improve the batteries capability and have a system able to use as little power as possible. In particular the transmission system is a power hungry part and its efficiency is then necessary to achieve longer life time.

In this work the focus is placed on how to reduce power consumption of the ADC of a transceiver while achieving a signal to noise ratio (SNR) and bandwidth in order for it to be used for wireless applications. The project is part of the ultra-portable devices (UPD) and is the continuing of the work of a former phd student at Lund University (Sweden) [29]. Previous versions were also made, the last one unfortunately presented some issues after fabrication, the aim of the work was to bring some improvements to the circuit and fix the issues so it will be possible to fabricate the chip again and measure it.

The first chapter offers an overview on continuous time $\Delta\Sigma$ introducing the basic working principles, the differences with their discrete time counterpart and how the design can be carried out at system level. The inherent benefits and drawbacks of the continuous time architecture will be discussed along with the most common techniques used to implement the modulators.

The second chapter will explain how the part of the modulator can be implemented and it will focus on low power architectures, for instance how to implement a loop filter with reduced number of opamps and low power realization for the internal quantizer. A discussion about the loop delay, which is one of the big limitations for continuous time modulators, is carried out here showing also how the different choices of DACs will affect the system. Also the degradation due to clock jitter is introduced and briefly discussed here.

In the third chapter the actual implementation of the system is shown, from the system level choices to the transistor level. Showing the implemented loop filter and why it was chosen for low power application, the internal quantizer and the feedback DAC.

The fourth chapter will present all the relevant results from the simulations done on the system, showing its performances focusing on the maximum SNR and stability mainly. In the last part of this chapter the methodology used to find and solve the issue with the fabricated circuit is explained and also the post layout simulations showing the issue is now fixed and the circuit is working are shown.

Chapter 1

$\Delta\Sigma$ Principles

1.1 $\Delta\Sigma$ Modulation

In this chapter the principles of $\Delta\Sigma$ modulators will be introduced, discussing why they are advantageous for ADC converters. A $\Delta\Sigma$ ADC basically works on two principles: oversampling and noise shaping. After a brief overview of these the first order modulator will be introduced and then higher order modulators. The last part of the chapter focuses on the continuous time implementation of the $\Delta\Sigma$ modulator, this is also the choice for a low power architecture hence it will be discussed with more detail and it will be seen in the other chapters as well.

1.1.1 Oversampling

ADCs can be divided in two major categories: Nyquist rate converters and oversampling converters. The first ones use a sampling frequency which ideally is the Nyquist sampling frequency(hence the name of this category), namely $f_s = 2f_b$ where f_b is the input signal bandwidth. The sampling frequency in practical applications will have to be higher than the ideal one as it requires an ideal filter to work. The second category of converters includes those converters that use an higher sampling frequency than the Nyquist one. The reason for it is that the quantization noise can be approximated as white noise [1, 5] thus its power spectral density(PSD) is uniform between $0Hz$ and $\frac{f_s}{2}$ and the power equal to $\frac{\Delta^2}{12}$, where Δ is the quantization step [28]. As it is shown in figure 1.1 the noise for frequencies higher than f_b can be filtered when $\frac{f_s}{2} > f_b$ so leaving a total noise power in the signal bandwidth of:

$$IBN = \frac{\Delta^2}{12} \cdot \frac{f_b}{f_s/2} = \frac{\Delta^2}{12} \cdot \frac{1}{OSR} \tag{1.1}$$

with $OSR = \frac{f_s/2}{f_b}$ being the oversampling ratio.

We see that increasing the sampling frequency, hence the OSR, the noise power over the signal bandwidth decreases, and for every doubling of the OSR the noise power is reduced by half, which means an increase of $3dB$ of the signal to quantization noise ratio, $SQNR$.

As an example we can see that to achieve a $14bit$ ADC using a $6bit$ quantizer and oversampling we need to increase the $SQNR$ of the quantizer by $8bit$. It can be

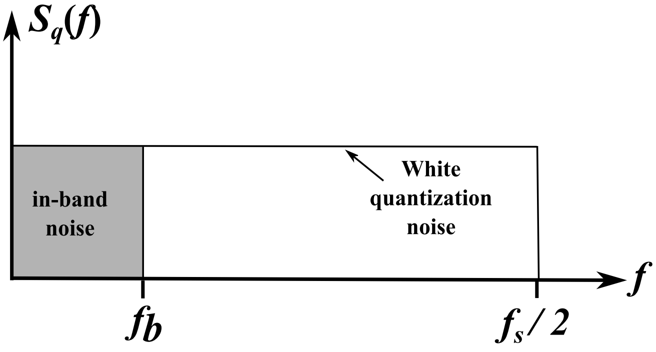


Figure 1.1: Effect of oversampling on the quantization noise

shown that every bit increases the $SQNR$ of about $6dB$ [28]. Hence in our example we need to increase the $SQNR$ by $48dB$ that means we can use $OSR = 2^{16}$ from what has being said earlier. For an audio signal with $f_b = 22kHz$ the sampling frequency will be $f_s = f_b \cdot OSR \approx 1.4GHz$.

We can here see the drawback of this technique: when the signal bandwidth is high oversampling, or the OSR has to be high, this technique might be impractical, due to the technology used, as it would require a too high sampling frequency.

1.1.2 Noise Shaping

Oversampling reduces the PSD of the quantization noise spreading its PSD over a wider range of frequency but to achieve high resolutions this might prove to be not practical. In order to make the oversampling more effective it's possible to filter the noise, for example using an high pass filter so the PSD will be lowered in the signal bandwidth and increased outside of it, figure 1.2.

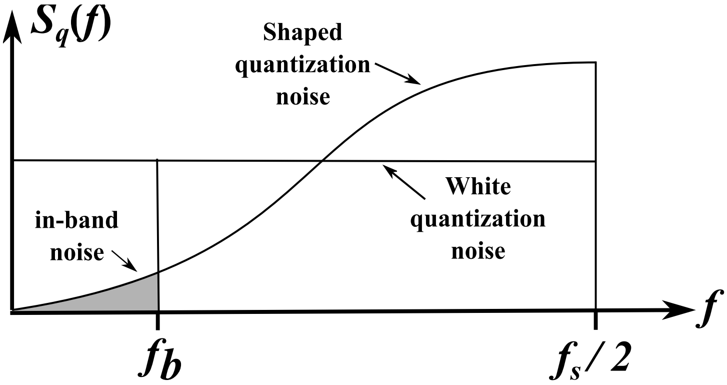


Figure 1.2: Effect of noise shaping, further reducing the noise power in the signal bandwidth

If we can achieve that, after filtering, as for the oversampling only case, the power of the noise will be decreased not only by the OSR factor but also from the filter attenuation. In this scenario we can observe that we can also use high order filters to suppress the PSD in the signal bandwidth. It's worth mention that the total noise power is always $\frac{\Delta^2}{12}$ but after the high pass filter the PSD is shaped so that the in band noise decreases but the out of band noise increases, so integrating till $f_s/2$ will always give the same power.

1.1.3 First order ΔΣ modulator

The system in figure 1.3 represents an ADC using a first order ΔΣ modulator. This is a common oversampled ADC and thanks to the loop filter(an integrator in this case) and the feedback it also presents a noise shaping effect. The system is commonly used with low resolution quantizers, the number of bits of the overall ADC is then increased by the loop filter, allowing the ΔΣ ADC to achieve the highest resolutions among other kinds of ADCs. As will be shown later the continuous counterpart of this system is also suitable for low power applications. The DAC after the quantizer is necessary to convert the feedback from digital to analog before feeding it back to the integrator.

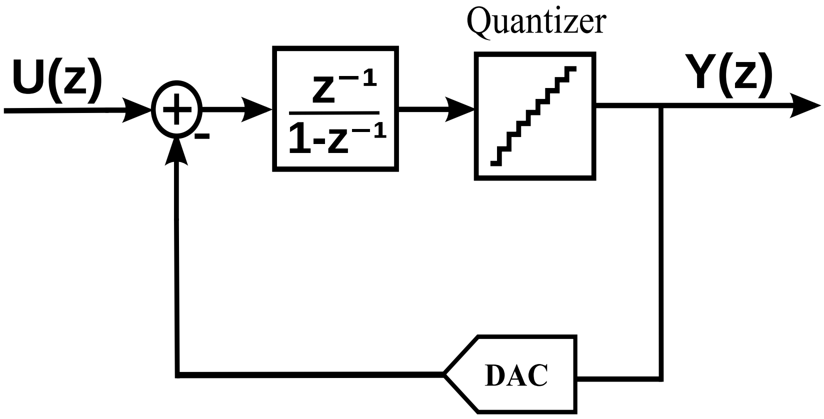


Figure 1.3: First order ΔΣ discrete time

The system in figure 1.3 is inherently non linear, as the internal ADC doesn't have a linear relationship between its input and its output, a simple linear model can be realized assuming the quantization noise to be white [1, 5]. In this assumption the model for the ADC become an adding block where the quantization noise is injected in the system. The liner model for the DAC is a simple unitary transfer function as its operation is to convert digital output into an analog signal. The resulting linearized system is shown in figure 1.4. From figure 1.4 we can make a simple analysis of the system and find that the output of the system in the Z-domain is:

$$Y(z) = z^{-1}U(z) + (1 - z^{-1})E_q(z). \tag{1.2}$$

We can observe that the transfer function from input to the output, called signal transfer function (STF) is a delay, hence at the output we will have a delayed copy of the input signal, it will be a digital copy. The noise transfer function, NTF, the one from quantization noise to the output is a first order differentiator. In other words the quantization noise at the output every instant is subtracted from the value of the preceding instant. We can see here see the effect of the noise shaping, leaving

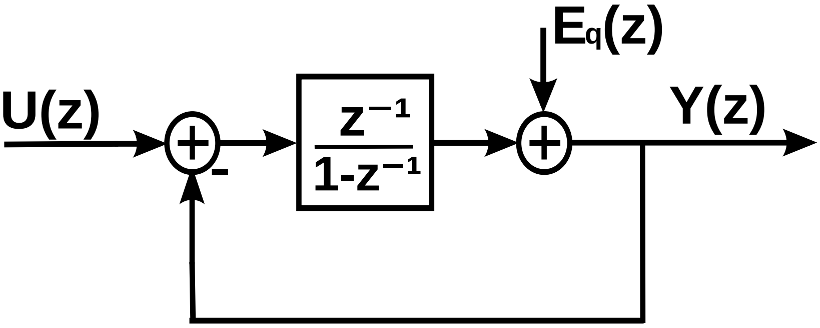


Figure 1.4: Linearized model for a first order ΔΣ discrete time

the input unaltered and attenuating the noise at low frequencies, ideally we have infinite attenuation at DC.

The analysis carried out is simplified and can't fully predict the stability of the system, which is of great interest for implementing the system. One reason is due to the fact that the quantizer gain is signal dependent and this is not modeled in the simple model made. Also if the quantizer is overloaded the quantization error can be higher than the quantization step. In this regard it's possible to have more complex models including these effects but extensive simulations will have to be carried out to test the stability range of the modulator.

Let's now analyze how the oversampling and the noise shaping of this system will affect the signal to quantization noise ratio (SQNR) of the converter. Said $S_q(f)$ the quantization noise PSD, that is constant in the assumption of white noise. We can then calculate the noise power in the signal bandwidth and hence the SQNR. The NTF in the Laplace domain can be obtained by substituting $z = e^{sT_s}$, and with $s = 2\pi f$ we obtain the filtered PSD of the quantization noise to be

$$S_e(f) = S_q(f)|NTF(f)|^2$$

from 1.2

$$S_e(f) = S_q(f)\left|1 - e^{-\frac{2\pi f}{f_s}}\right|^2 = S_q(f)\left(2sin\left(\pi\frac{f}{f_s}\right)\right)^2$$

Recalling that $S_q(f) = \frac{\Delta^2}{12}\frac{1}{f_s}$ and $sin\left(\pi\frac{f}{f_s}\right) \approx \pi\frac{f}{f_s}$ for small values of $\frac{f}{f_s}$ the in band noise will then be

$$\begin{aligned} IBN =&2\int_0^{f_b} S_e(f)df = 2\int_0^{f_b} \frac{\Delta^2}{12}\frac{1}{f_s}\left(\pi\frac{f}{f_s}\right)^2 df \\ =&\frac{\Delta^2}{12}\frac{\pi^2}{3}\frac{1}{OSR^3} \end{aligned} \tag{1.3}$$

We can already compare (1.3) with (1.1) and see that when we combine a simple first order noise shaping function with oversampling the noise power is reduced by a factor of OSR^3 compared to OSR when having only oversampling. In terms of SQNR this will translate in $8dB$ gain for each OSR doubling compared to the $3dB$ for the oversampling only case. Thus combining oversampling with noise shaping will enhance the effects of the two techniques.

1.1.4 Higher order ΔΣ modulator

To further improve the benefits from the noise shaping we can implement an higher order differentiator compared to the one of the NTF in (1.2). This will provide greater attenuation for the IBN thus enhancing the benefits from the noise shaping technique. To do so the system in figure 1.4 can be generalized to the one in figure 1.5, where $H_0(z)$ and $H_1(z)$ are generic Z-domain transfer function. For this system we still have

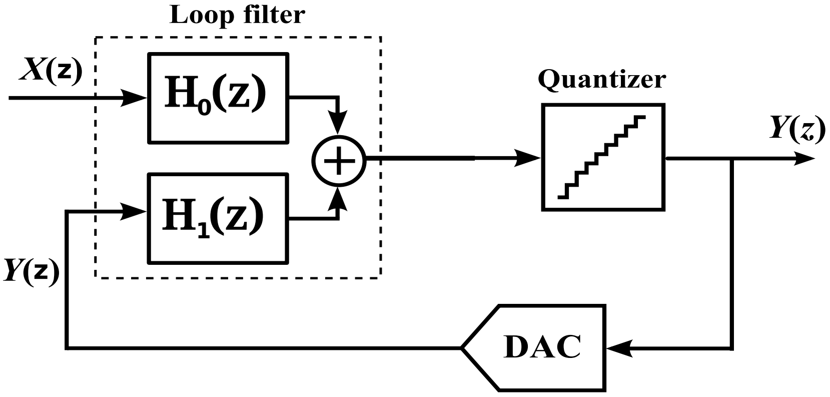


Figure 1.5: Linearized model for generic ΔΣ discrete time

$$Y(z) = STF(z)U(z) + NTF(z)E_q(z)$$

and it can be easily shown that

$$\begin{aligned} STF(z) =& \frac{H_0(z)}{1 - H_1(z)} \\ NTF(z) =& \frac{1}{1 - H_1(z)} \end{aligned}$$

Hence it is possible to implement different kind of NTF by changing $H_1(z)$.

We can see from figure 1.6 the drawback of increasing the order of the NTF. Increasing the in band attenuation brings an increase in the out of band gain, which rapidly increases with the order of the filter. This gain eventually leads to saturation in the quantizer and this effect will make the system instable. A trade off between the NTF order, the oversampling and the number of quantizer of the internal quantizer has to be found to ensure the modulator will be stable for the given input range.

About the choice of the quantizer historically the use of a one bit quantizer has been common. This is due to the fact that realizing the ADC doesn't require much effort since all of its non idealities will be injected in the loop filter where the quantization noise is injected, thus they will be fully modulated by the feedback and won't represent a critical issue in the system. On the other hand when the output is fed back into the loop we need to convert it to the analog domain using a DAC, but at this point of the loop filter all the DAC non idealities will be injected directly into the input, hence they won't be modulated at all. For this reason the DAC linearity has to be as good as the overall ADC. It's easy to understand now why choosing a

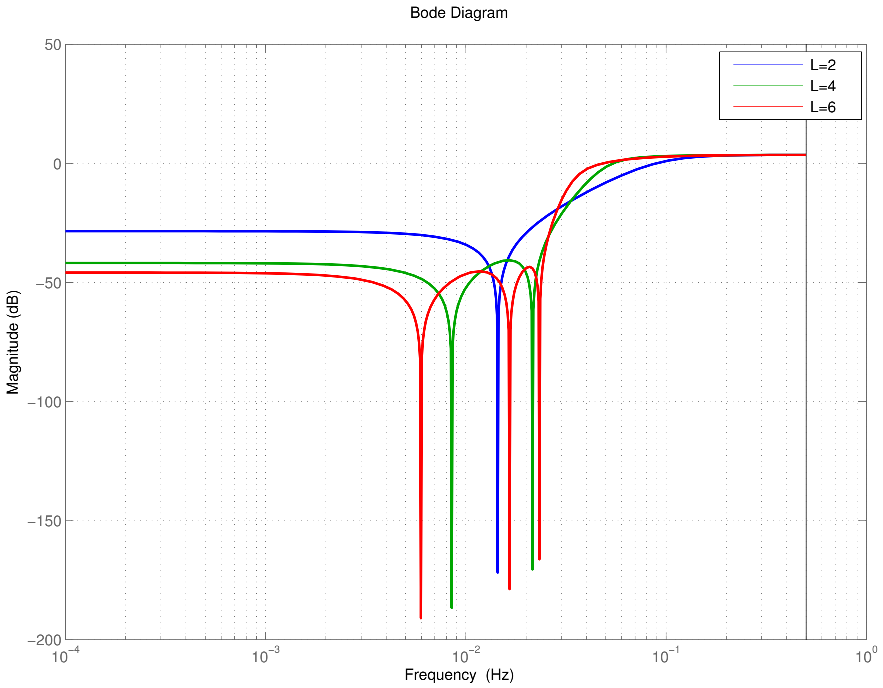


Figure 1.6: Plot of different NTFs showing how the out of band gain increases more rapidly increasing the order

one bit ADC is convenient, having only two different values the DAC output will always be linear, hence making it easier to implement the DAC without affecting the system performances.

Even though using a multi bit internal quantizer has some advantages. First of all increasing the resolution reduces the quantization step, reducing the quantization noise power directly. It was also shown that with multibit quantization the modulator is more stable as the quantizer gain can be approximated to be unity gain, and it makes the assumption of white quantization noise more valid [11]. Again there's a trade off between these advantages from a multibit quantizer and the DAC non linearity.

1.2 Continuous time ΔΣ modulators

So far all the systems shown are discrete time systems, meaning that the input is sampled before entering the loop filter, and hence an anti aliasing (AA) filter is required before the sampling operation and the loop filter is discrete time. DT ΔΣ converters have been popular in the last two decades due to the fact that their circuit realization is highly insensible to process variations and it's also quite straight forward to implement the converter from the math behind it. In fact the realization of the loop filter for DT ΔΣ is implemented using switched capacitor technique which make the loop filter realization simple and give high linearity.

Another way to implement the loop filter is to use a continuous time (CT) transfer function, figure 1.7, and actually the first implemented ΔΣ was a CT system [14]. Recently the CT counterpart has received increasing attention due to some inherent benefits:

- an AA filter is implemented by the ADC and it can ease the requirements of the AA filter at the input or even make it not needed anymore
- CT is more suitable for low power applications, as the settling time of the amplifier is less critical.

We are going to briefly review the major differences between the two topologies and their advantages and drawbacks, one methodology for designing a CT modulator will be also shown. Before doing so another way of representing the linear model of the modulator is given, this is useful to understand how the AA filter and single opamp network works in a CT modulator. This representation is shown in figure 1.8, here the paths for input and feedback signal are split and $FF(s)$ can be seen as a prefilter of the input signal. Since the input is a continuous signal while the output is a discrete time one is not immediate to define a STF for this system as was done for the DT system. How the NTF is affected will be discussed when the conversion between DT and CT is discussed. For the system in figure 1.8 the output can be written as:

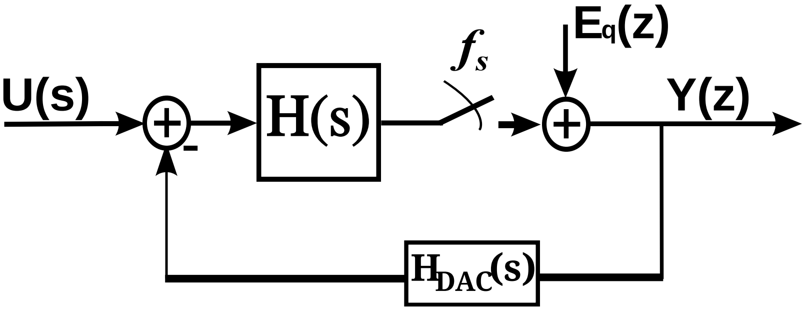


Figure 1.7: Linearized model for generic ΔΣ continuous time

$$\begin{aligned} Y(z) &= NTF(z)\,[FF(s)X(s)]^* + NTF(z)E_q(z) \\ &= \left[NTF(e^{sT_s})FF(s)X(s)\right]^* + NTF(z)E_q(z) \end{aligned}$$

where $E_q(z)$ is the quantization error and the * operator is the periodic repetition of the spectrum after sampling, defined in [18]. Hence for a CT modulator the STF can be found as:

$$STF(s) = NTF(e^{sT_s})FF(s). \tag{1.4}$$

As already found for DT also in the CT the STF depends on the NTF.

1.2.1 Sampling and quantization

In figure 1.7 a linear model for a CT ΔΣ is depicted, comparing it with its DT counterpart in figure 1.5 we see that the sampling operation take place inside the

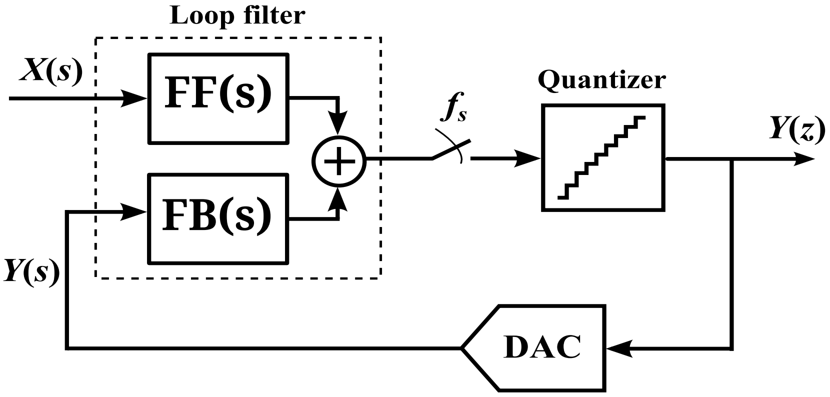


Figure 1.8: Linearized model for generic ΔΣ continuous time modulator showing two different transfer function for input and feedback signal

loop for the CT system. This is of course the case as said before the loop filter is of a continuous time kind. Bringing the sampling operation inside the loop filter is what provide the CT modulator with an inherent AA filter and thus represents one of the big advantages of this architecture, more of the AA filter will be discussed in next session 1.2.3.

In practical implementations the sampling will be inside the quantizer, this means that all the non idealities introduced by the sampling and hold block will see the maximum noise suppression, in fact they are injected at the same point with the quantization noise. In the DT time instead the sample and hold block non idealities are all introduced at the input, therefore not being modulated at all.

Having a CT system means though that the whole waveform of the signal contributes to the output, hence the quantizer, and the DAC also, should be ideally fast and don't introduce any delay in the feedback loop as the analog copy of the output is needed immediately for the feedback. This was not the case for DT systems that are able to tolerate delays introduced by the quantizer and DAC operations. The delay from the quantizer is critical then in CT implementations and can lead to instability of the system, this is the case when using a successive approximation ADC, which is desired for low power applications. Loop delay is a major issue with this design and it will be discussed in detail in next chapter where techniques to prevent instability from loop delay are given.

1.2.2 Design of the loop filter for a CT ΔΣ

In figure 1.7 a block representing the DAC transfer function, $H_{DAC}(s)$, was added when comparing it with figure 1.5. The reason for it is that for a DT loop filter only the signal at sampled times matters while all the transients in between two sample don't influence the output, so as log as the quantizer and the DAC are fast enough to provide the right value before the next sampling period the transition of the signal doesn't matter. For a CT filter this is not the case as the loop filter will always be filtering its inputs, here the transition of the signal in between two sampling periods will influence the filter output, hence we must take it into account when analyzing the system.

To design the loop filter of a CT ΔΣ it is common to start with a DT system, as in the past years this was the most common nowadays several tools can be found to do so, which makes the process of designing a DT ΔΣ more easy. So a popular approach is to synthesize the DT modulator and then converting it into the CT domain [4, 10, 11, 13, 23, 34].

It was shown that one method to convert a DT modulator to CT is the impulse invariant transformation [2, 34]. This approach is based on the observation that the system in figure 1.7 become a DT system after the quantization, therefore the method focus on imposing the open loop transfer function in both DT and CT cases equal at the sampling time. This translate in the following equations:

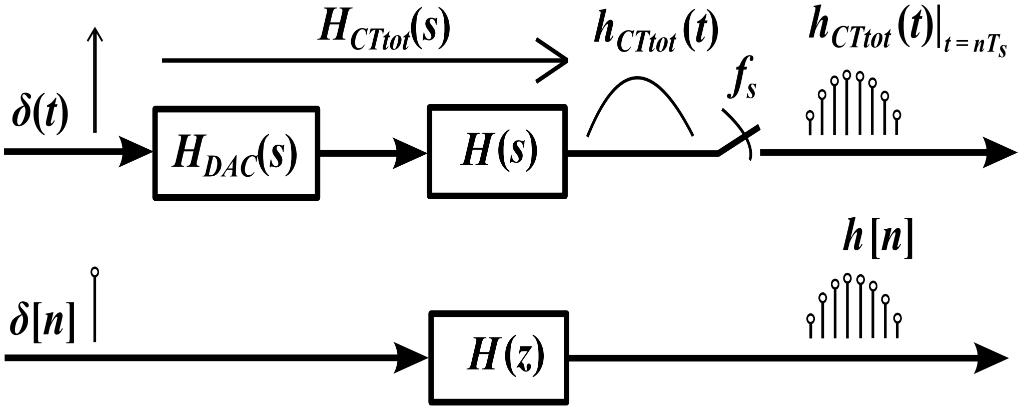


Figure 1.9: Impulse responses from the output of the quantizer to its input for both CT and DT

$$\mathcal{Z}^{-1}\{H(z)\} = \mathcal{L}^{-1}\{H_{DAC}(s)H(s)\}\big|_{t=nT_s}$$

applying the Laplace and $\mathcal{Z}$ transform inverse operators we obtain:

$$h(n) = \left[H_{DAC}(s) * H(s)\right]\big|_{t=nT_s} = \int_{-\infty}^{+\infty} H_{DAC}(\tau)h(t-\tau)d\tau \qquad (1.5)$$

The time domain equation can be now used to design a proper loop filter function, given the DT loop filter transfer function and the DAC transfer function [21]. As said there are computer tools tools for designing a DT modulator, given a NTF and a given topology for the modulator they will automatically calculate the required coefficients to be implemented. Using Schereier's toolbox [33], a matlab tool, it is possible to do so and also to apply the impulse invariant transformation and from a DT time modulator automatically obtain the CT coefficients.

1.2.3 Inherent anti aliasing filter for CT ΔΣ

To show the origin of the AA filter into CT modulators we will use as example as second order modulator figure 1.10, for simplicity we assume the quantizer and the DAC to be delay free and analyze the time domain function of the system. The

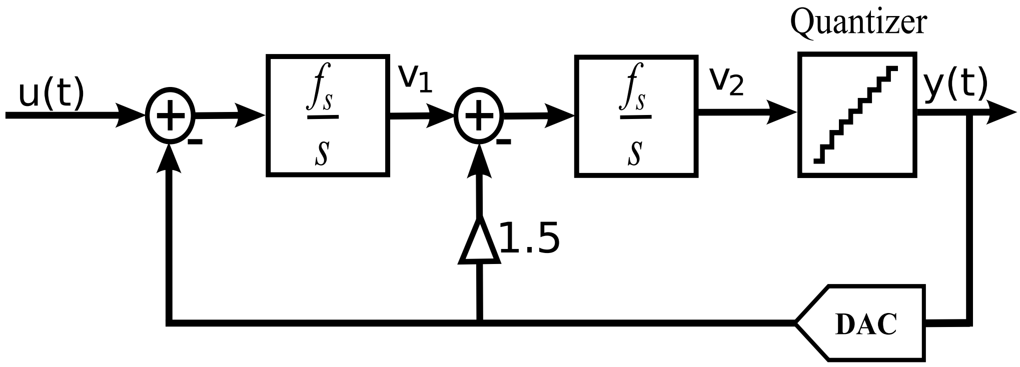


Figure 1.10: Scheme of a CT second order modulator

output of the first integrator for times $n \geq t < n+1$ can be easily obtained from:

$$v_1(t) = v_1(nT_S) + \int_{nT_S}^{t} (u(\tau) - y_{ct}(\tau))\, d\tau \tag{1.6}$$

here y_{ct} is the continuous time version of the output. Since DAC and quantizer are ideal but still clocked, between a generic sample time nT_S and its successive $(n+1)T_S$ the output of the DAC is constant with value $y_{ct}(n)$. Equation (1.6) becomes for $t = (n+1)T_S$

$$v_1\left((n+1)T_S\right) = v_1(nT_S) + y_c(n) + \int_{nT_S}^{(n+1)T_S} u(\tau)d\tau \tag{1.7}$$

Here $\int_{nT_S}^{(n+1)T_S} (u(\tau))\, d\tau$ is called boxcar integration [34] and it can be seen as the convolution of the input signal with a non casual filter. To show that we apply the transformation $\tau = nT_S - \tau$ and consider. We obtain then:

$$-\int_{0}^{-T_S} u(nT_S - \tau)d\tau = \int_{-T_S}^{0} u(nT_S - \tau)d\tau = (h_1 * u)(nT_S)$$

where we used the definition of convolution and

$$h_1(t) = \begin{cases} 1, & -T_S \leq t \leq 0 \\ 0, & \text{otherwise.} \end{cases} \tag{1.8}$$

As we said this filter is non casual, it's non zero for negative times, we will see soon that this is not the whole filter applied to the input signal, and the whole filter will be casual. It can be shown [28] with analog reasoning that the output from the second amplifier is similar to (1.7)

$$v_2\left((n+1)T_S\right) = v_2(nT_S) + v_1(nT_S) + \int_{-T_S}^{0} (\tau+1)u(nT_S - \tau)d\tau - 2y(n)$$

in this case the integral part is the convolution $(h_2 * u)(nT_S)$ and

$$h_2(t) = \begin{cases} t+1, & -T_S \leq t \leq 0 \\ 0, & \text{otherwise.} \end{cases} \tag{1.9}$$

When putting together (1.8)(1.9) the result is to filter the input with the signal:

$$h(t) = \begin{cases} t, & 0 \leq t \leq T_S \\ 2 - t, & T_S \leq t \leq 2T_S \\ 0, & \text{otherwise.} \end{cases}$$

as expected this is a casual filter, and has a triangular shape, figure (1.11). To see

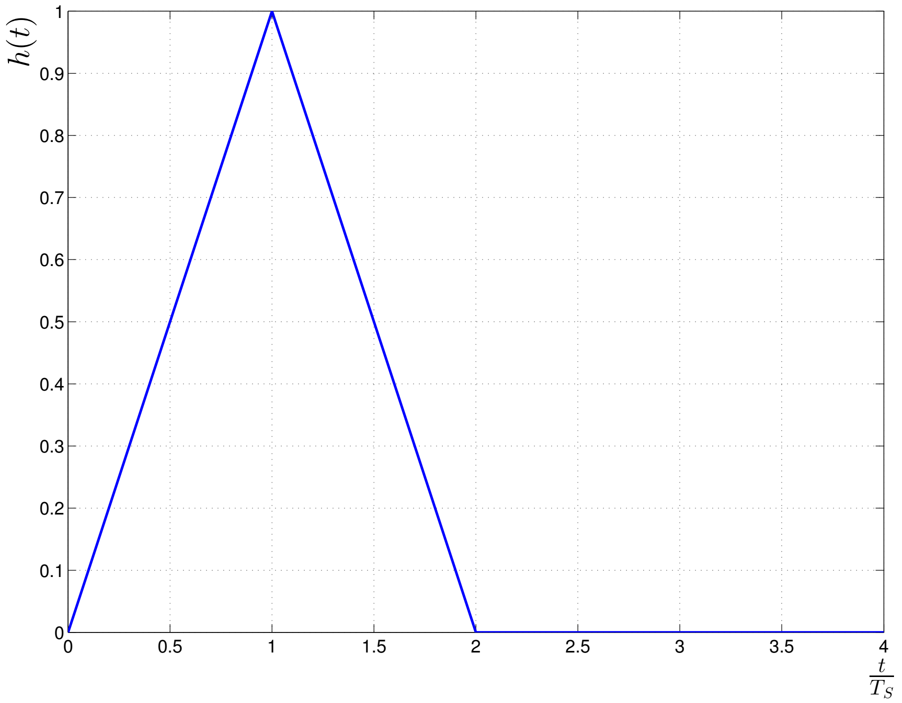


Figure 1.11: Shape of the filter applied to the input signal in CT modulators

the anti aliasing property of the filter we calculate its Laplace transform, assuming $T_S = 1$

$$H(s) = \left(\frac{1 - e^{-s}}{s}\right)^2.$$

Evaluating its magnitude in the frequency domain allows to calculate the STF of the system as found in [28] this is given by:

$$|STF(f)| = \left|\frac{sin(\pi f)}{\pi f}\right|^2$$

this transfer function has zeros at the multiple frequencies of the sampling frequencies, it has indeed an inherent anti aliasing filter, in figure 1.12 the magnitude of AA filter is plotted. Also the order of the zeros is the same of the order of the modulator, in this example a second order modulator give a AA filter with 2nd order zeros.

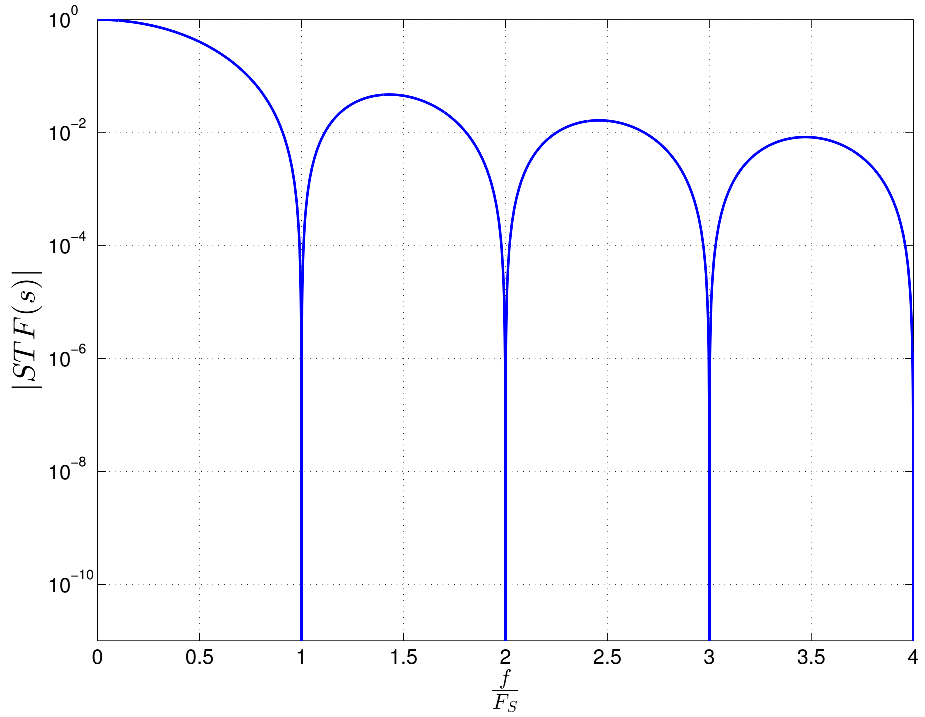


Figure 1.12: Magnitude of STF for the second order modulator showing its AA property

1.2.4 CT modulators for low power applications

Another major reason for the CT ΔΣ to be more popular nowadays is due to low power applications. The most power consuming parts in a ΔΣ ADC are the active elements, and in particular the opamps of the loop filter. In fact the filter is usually implemented using active RC integrators, in this configuration it's in fact possible to achieve good linearity, this is unfortunately dependent on the linearity of the differential pair in the input stage of the amplifier [17], hence the linearity can be increased using more current for biasing the differential pair.

When it comes to DT implementation the settling time of the amplifier must be short enough to guarantee proper operation, this is a big limitation to the sampling frequency, hence the clock of the system as it will require amplifiers with high bandwidth. In the CT architecture the filter works on analog signals and this drastically relaxes the speed requirements of the amplifiers, that in theory can be 10 times lower, but in practical realizations will be around 5 [6, 37]. In low power application it is then desirable to use CT modulators as their lower band requirements on the opamps will lead to lower power consumption for the same clock frequency. In a low voltage supply case the DT will also suffer from the R_{on} resistance of the switches which increases with the overdrive, this is completely avoided in CT designs as there is no need for switches on the signal path in the loop filter.

In the last few years CT ΔΣ modulators have been used to achieve low power consumption in ADCs circuits, for example in [20] a modulator achieving a maximum SNR of $91dB$ and consuming only $230\mu W$ was made. In [26] and [38] were presented modulators achieving $89dB$ and $62dB$ of peak SNR and consuming $122\mu W$ and $890\mu W$ respectively.

We will now focus on CT modulators as the project aim to build an ultra low power (less than $1mW$ power consumption) ADC and thus a CT architecture was chosen.

Chapter 2

CT $\Delta\Sigma$ modulators implementations

In this chapter the most common implementations of CT modulators are presented, the issues specif to CT design are also shown. We will also discuss single opamp networks that can be used to implement the loop filter with a reduced number of amplifiers, hence reducing the power consumption of the system.

2.1 Loop delay

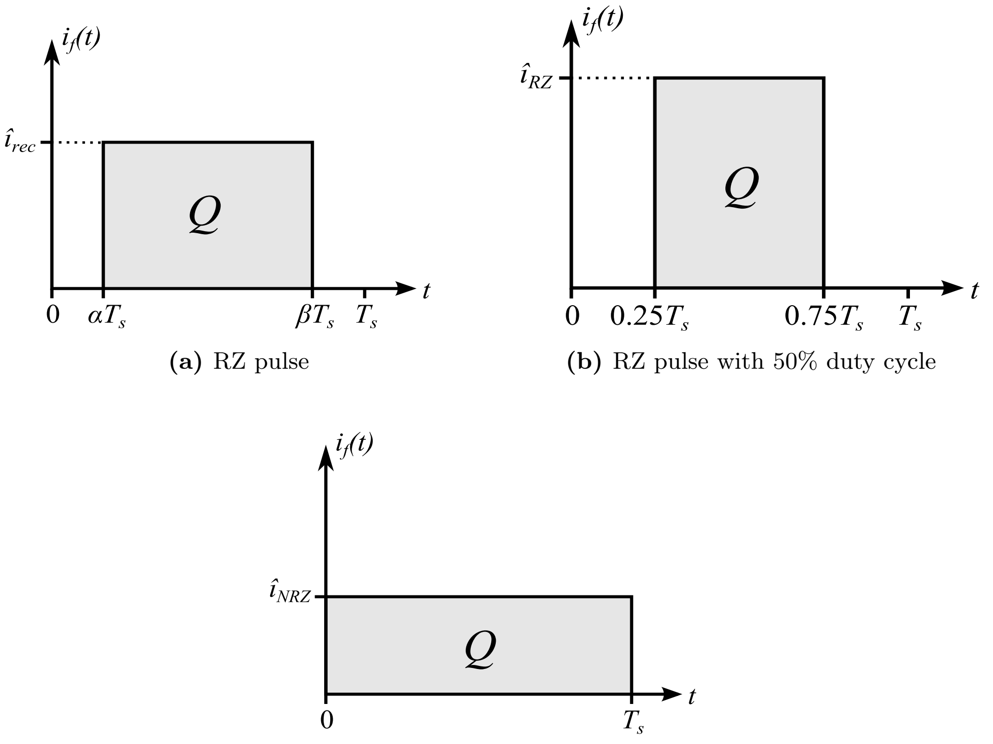
As discussed in previous chapter real quantizer and DAC require a certain amount of time to sample and quantize the output of the loop filter and feed it back. This delay can lead to severe performance degradation and instability in the modulator. On a side note on DT this is not an issue as quantizer and DAC have time to settle their outputs to the right value before the next sampling period, going to CT this becomes a major issue for the system.

2.1.1 DAC pulses

Since the effects of the loop delay are seen when the signal of the DAC returns to the input stage of the loop filter, the loop delay is seen as a delay in the DAC pulse, to understand then how the delay affects the system we will see its effects on the DAC's output waveforms.

The most common DAC pulses are rectangular pulses [21, 34], figure 2.1, these are the simplest waveform to implement. As shown there are two major cases for these pulses: return to zero (RZ) and non return to zero (NRZ) waveforms. The names are self explanatory, the first waveform is a pulse that will always go back to zero before the end of the sampling period while NRZ will update its value at most once per sampling period. Hence the RZ waveform will transition two times every cycle while the NRZ will transition at most once, in fact the output of the DAC might be the same for two or more cycles and its output don't need to change, as it is already at the correct output level. It is easy to see that for a RZ must hold that $\beta - \alpha < 1$ and also for a NRZ $\alpha = 0$ and $\beta = 1$.

Rectangular pulses are easy to realize, figure 2.2 depicts a switched resistor architecture, controlling the switches properly allow to have NRZ or RZ pulses, the



(a) RZ pulse

(b) RZ pulse with 50% duty cycle

(c) NRZ pulse

Figure 2.1: Rectangular DAC pulses

latter will require more circuitry to also open the switches before the next sampling period, while NRZ only needs to update the switches configurations when the input of the DAC changes.

From figure 2.1 we can calculate the Laplace transform and hence the DAC's transfer function:

$$H_{DAC}(s) = i_d \frac{e^{-\alpha T_S s} - e^{-\beta T_S s}}{s}. \tag{2.1}$$

Equation (2.1) can now be used with in the invariant impulse invariant transformation in (1.5) for calculating the coefficients of the CT ΔΣ modulator.

We will now analyze how a loop delay affects those two kinds of waveforms. In figure 2.3 a comparison with and without loop delay for RZ and NRZ pulses is shown.

As expected when a loop delay t_d is introduced the two shapes are shifted, the RZ one is still inside the correct sampling period as long as $t_d < T_S(1-\beta)$, hence $T_S(1-\beta)$ is the margin the RZ pulses have before entering the next sampling period. As oppose to it a NRZ pulse will certainly go into the next sampling period, in fact $\beta T_S = T_S$ in this case, hence the margin is zero.
This suggests that RZ DAC are less sensitive to loop delay, while NRZ are always affected by it.

As said the most significant effects of loop delay can be seen from these considerations on the DAC pulses:

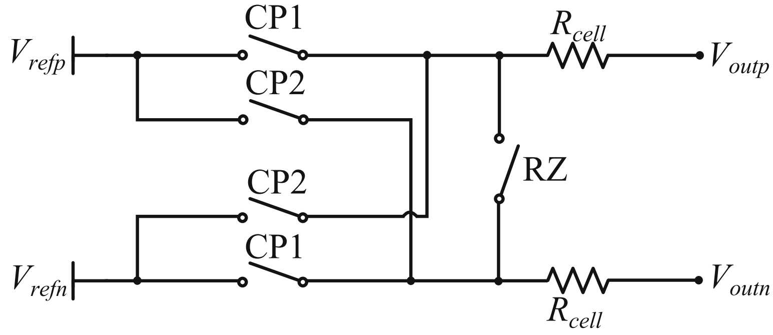


Figure 2.2: DAC resistive cell used to implement rectangular shaped pulses

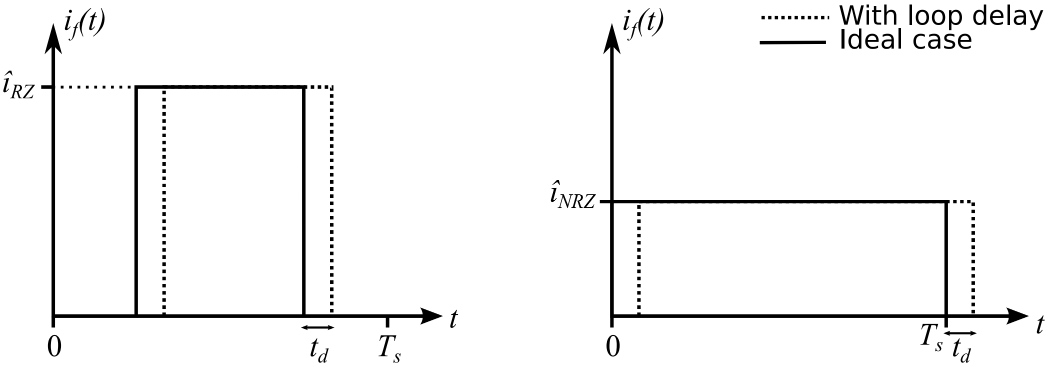


Figure 2.3: Rectangular DAC pulses affected by loop delay

- Coefficients variation: when a pulse don't exceed into the next sampling period, thus only for RZ, we can see the delay t_d affects the coefficients of the DAC so α and β will become $\alpha + \tau_d$ and $\beta + \tau_d$ where $\tau_d = \frac{t_d}{T_S}$ is the normalized loop delay. This change will affect the transfer function (2.1). And then (1.5), the transformation will then yield the coefficients for a CT system that is different from the ones calculated without considering loop delay. This affect the NTF of the system and will increase the quantization noise degrading the performance of the modulator. Since the NTF also affect the stability of the system it can affect the stability of the modulator as well. To prevent this to happen it's possible to determine the loop delay of the system and introducing it into the impulse invariant transformation calculate the coefficients for the CT taking into account of the loop delay. One problem that arise with this approach is that the quantizer will take more time to process small inputs and thus the loop delay will not be constant.

- Increased modulator order: when the feedback pulse is pushed into the next sampling period, note that this will always happen in case of NRZ DACs and loop delay, the order of the modulator will result increased by one. For example a NRZ waveform will have $\alpha = 0$ and $\beta = 1$, with the loop delay this will become $\alpha = \tau_d$ and $\beta = 1 + \tau_d$. The pulse can be seen as two different pulses, one with $\alpha_1 = \tau_d$ and $\beta_1 = 1$, hence going from t_d to T_S, while the second part is a pulse in the next sampling period with $\alpha_2 = 0$ and $\beta_2 = t_d$. It was shown in [21] that this effect increases the numerator of the DAC transfer function. If we again apply the impulse invariant transformation we will obtain one more degree of freedom due to the increased order of the DAC transfer function. Again the effect of the loop delay alter the NTF coefficients and can lead to an unstable modulator if it's large enough.

When designing a low power system it's becoming always more common to use successive approximation register ADC, as it is the most energy efficient architecture for a quantizer. This kind of ADC will take several clock cycles to elaborate its input, hence its delay can be quite relevant, especially in a CT implementation. We will discuss this issue when talking about the quantizer.

2.1.2 Loop delay compensation

As seen the presence of a loop delay is detrimental to the performance of the system and when the order of the modulator is increased it can also lead easily to instability.One technique to prevent its effects is proposed in [6] and consists in adding one more feedback path from the output of the quantizer to the output of the last integrator. An intuitive explanation can be given observing that if the order of the modulator increases it will give one more degree of freedom in (1.5), to restore system's controllability and being able to apply the transformation a new path that adds another coefficient is inserted.

We will see an example of this on a second order modulator in figure 2.4 a standard 2nd order modulator with a CIFB structure is shown. The system is the

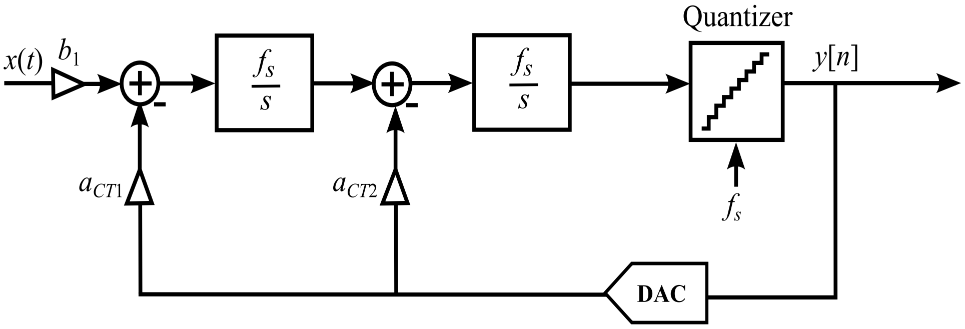


Figure 2.4: Second order CIFB structure

CT equivalent of a DT modulator with $NTF = (1 - z^{-1})^2$, where a_{CT1} and a_{CT2} are the coefficients of the global feedback paths, found using the impulse invariant transformation from the DT CIFB structure.

In figure 2.5 the same system is depicted but with the added compensation path. To take into account the loop delay, and so the compensation path, the continuous time coefficients found from the impulse invariant transformation need to be modified, for example in [21] it has been shown that for the system depicted in figure 2.5 the

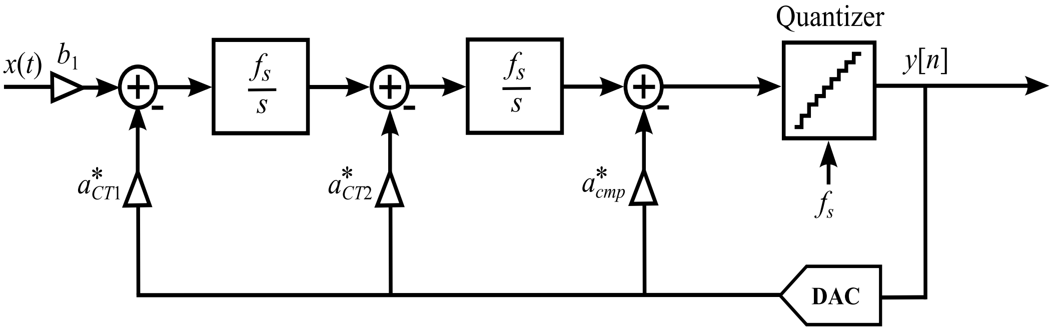


Figure 2.5: Compensation DAC path in a CIFB structure

coefficients for CT will be:

$$\begin{cases} a^*_{CT1} & = a_{CT1} \\ a^*_{CT2} & = a_{CT1}\tau_d + a_{CT2} \\ a^*_{cmp} & = \frac{a_{CT1}\tau_d^2}{2} + a_{CT2}\tau_d \end{cases} \tag{2.2}$$

Taking into account the loop delay hence means also a rescaling of the coefficients to implement the desired NTF. From equation (2.2) it is possible to find the new coefficients when the loop delay is known, in this case a worst case estimation is needed. Otherwise it is also possible to see how the loop delay changes varying a_{cmp} coefficient, this latter method can be useful when a certain gain factor is desired on the compensation path for implementation reasons, as will be the case for the project realized for this thesis. If a_{cmp} is imposed we will need to use a certain amount of loop delay, given by (2.2), that has to be fixed and also it must ensure proper settling of the ADC and the DAC in its duration.

In [3] it was also shown which paths are the most sensitive to loop delay. The first DAC path, going from the quantizer output to the input of the system is the least sensitive while the compensation path is the most sensitive. After seeing that the RZ pulses are more robust against loop delay than the NRZ pulses this will give an insight how to implement the DAC on the different paths. After this considerations it will be reasonable to use a RZ on the compensation path.

2.1.3 Jitter sensitivity in CT $\Delta\Sigma$

To conclude the analysis on DAC pulses for CT $\Delta\Sigma$ we need to discuss about clock jitter errors, this is considered probably the biggest drawback of CT architecture, this is due again to the nature of the loop filter that will integrate the whole DAC waveform thus the jitter errors in the DAC pulses will be integrated and will increase the noise. Jitter errors will in fact change the position of the feedback pulse by a certain amount, this will result in a different value of charge transferred into the loop filter. Opposed to the loop delay here the first feedback path is the most affected by errors because they are injected directly to the input where they won't see any noise shaping. The other paths will instead see some degree of noise shaping, depending on their position, and hence will be less critical for DAC timing errors. Jitter errors on the sampling operation will also be attenuated by the NTF as the sampling is done in the quantizer and hence inside the loop filter that's why the study of jitter errors focuses on DAC pulses, they are the relevant source of errors due to clock jitter.

Clock jitter depends on the clock source used, ideally the clock signal is periodic and the rising edges of the clock happens always at multiple times of T_S, considering the sampling operation happens with rising edge. In real implementations the edges of the clock will happen at different times from the ideal clock. This effect can be modeled in two ways: independent jitter (white jitter also) and accumulated jitter.

In presence of white jitter the sampling instants will be:

$$t_n = nT_S + t_{jn} \tag{2.3}$$

where n is the n-th clock cycle, and t_{jn} are independent identically distributed (iid) random variables representing the time variation from the ideal edge. In this case the variation of a generic n-th edge time will be independent from the other variations in the previous instants, hence why it's called white jitter.

It was shown in [9, 24] that a more realistic model for the clock jitter needs to take into account of the variations in the previous instants, the jittered sampling time can be expressed as:

$$t_n = nT_S + \sum_{i=0}^{n} t_{ji}. \tag{2.4}$$

Here the n-th sampling time depends also on the history of the clock waveform and depends on the sum of the iid variables on previous instants.

It was shown that jitter errors on the DACs, hence error on the charge transferred to the loop filter can be modeled as additive amplitude errors on an ideal DAC [7, 16, 35]. The DAC model become similar to the ADC ones, when the jitter error has the same model of the quantization error on an ADC, figure 2.6. The system

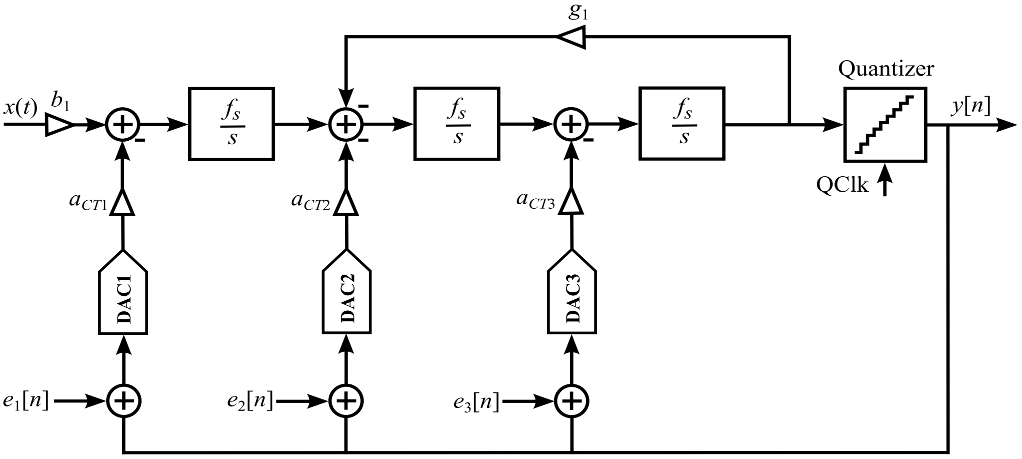


Figure 2.6: DAC model including the additive noise due to jitter errors

in figure 2.6 also better explain why the first path of the DAC is more sensitive on jitter errors, in fact the errors sum directly on the input and will be modulated by the STF, meaning they will see unity gain in the signal bandwidth. On the other hand the feedback paths closer to the quantizer see at least a first order modulation, hence their non idealities will have some meaning of noise shaping, even though jitter errors should be modeled for other paths than the first one as their contribution might not be negligible overall. This is also true for all non idealities in the first DAC path, for example the linearity of this block will set the linearity of the whole system, that's why using one bit ADC was so popular.

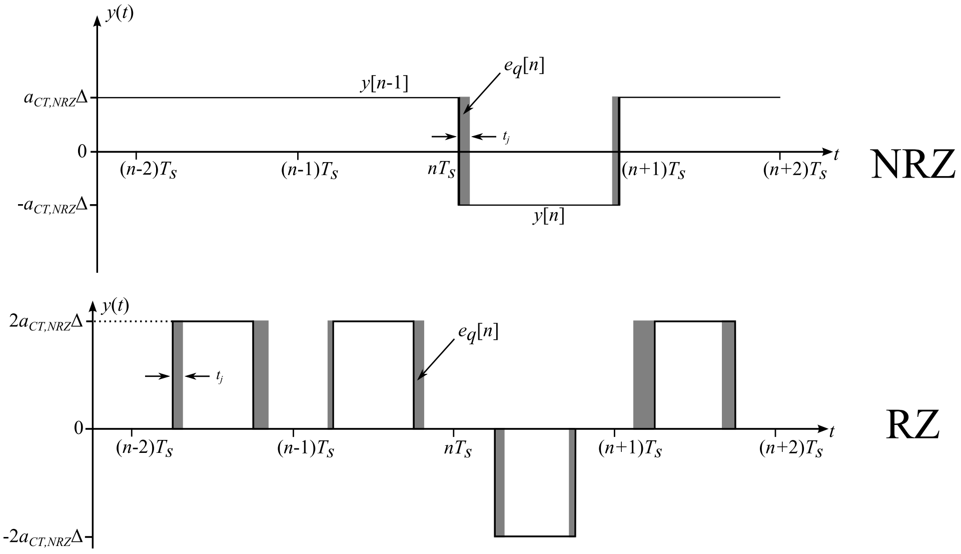


Figure 2.7: DAC pulses affected by clock jitter

Now we analyze how RZ and NRZ are affected by clock jitter, assuming for now a one bit DAC, hence with only two different output levels. From figure 2.1 it's possible to see that the jitter error will affect the position of the transition in the DAC output, hence changing the area of the rectangle, and thus the charge transferred to the loop filter. The variation of the charge transferred will be dependent on the height of the waveform and the difference between the ideal clock and the jittered one. Said that $a_{CT,NRZ}$ and $a_{CT,RZ}$ are the feedback path coefficients for a NRZ and RZ DAC, $i_{d,NRZ}$, $i_{d,RZ}$ are the current values of the two DACs and the difference between ideal clock and real clock is modeled by t_j then

$$dQ = 2a_{CT,NRZ} \cdot i_{p,NRZ} \cdot t_j \tag{2.5}$$

is the charge difference between jittered clock and non jittered clock for a NRZ DAC, as can be seen in figure 2.7.

The two factor comes from the two levels that are $\pm a_{CT,NRZ} i_{p,NRZ}$. Also (2.5) is valid for a RZ DAC for an equivalent system. To see this we need to take into account that the charge transfered from the DAC to the loop filter for two equivalent systems, i.e. implementing the same NTF, must be the same. Then for a 50% duty cycle RZ pulse it must be $i_{d,RZ} = 2i_{d,NRZ}$, and since the transition goes from $i_{d,RZ}$ to zero equation (2.5) is still valid for a RZ DAC.

In figure 2.7 we see a graphical representation of the jitter affecting the DAC pulses, we note from it that a RZ DAC is forced to transition two times every cycle, hence both of its edges are affected by the jitter noise. The charge error will be increased then compared to the NRZ case, this reflects in a bigger additive error on the DAC. Using the additive error modeling and white jitter noise for simplicity it was shown [21] that for an one bit DAC using rectangular pulses in the first path the IBN due to jitter is given by:

$$IBN_j = \frac{A_f}{OSR}\left(\frac{a_{CT1}}{b_1}\frac{\Delta\sigma_j}{T_S}\right)^2$$

$\Delta\sigma_j$ being the standard deviation of the jitter white noise and A_f the activity factor of the pulse, that indicates the number of transition in one cycle. As already pointed out for RZ it will be $A_f = 2$ and for NRZ $A_f \leq 1$. In [34] it was found that for a large input and single bit NRZ DAC $A_f \approx 0.7$. It is then possible to calculate the ratio between RZ and NRZ in band noise:

$$\frac{IBN_{RZ}}{IBN_{NRZ}} = \frac{A_{f,RZ}}{A_{f,NRZ}} = 2.8 \sim 4.5dB. \tag{2.6}$$

Which confirms that RZ DACs are more sensitive to jitter errors, and the IBN due to their contribution is about $4.5dB$ higher than a NRZ DAC, hence it is advisable to use a NRZ DAC in the first feedback path of the modulator.

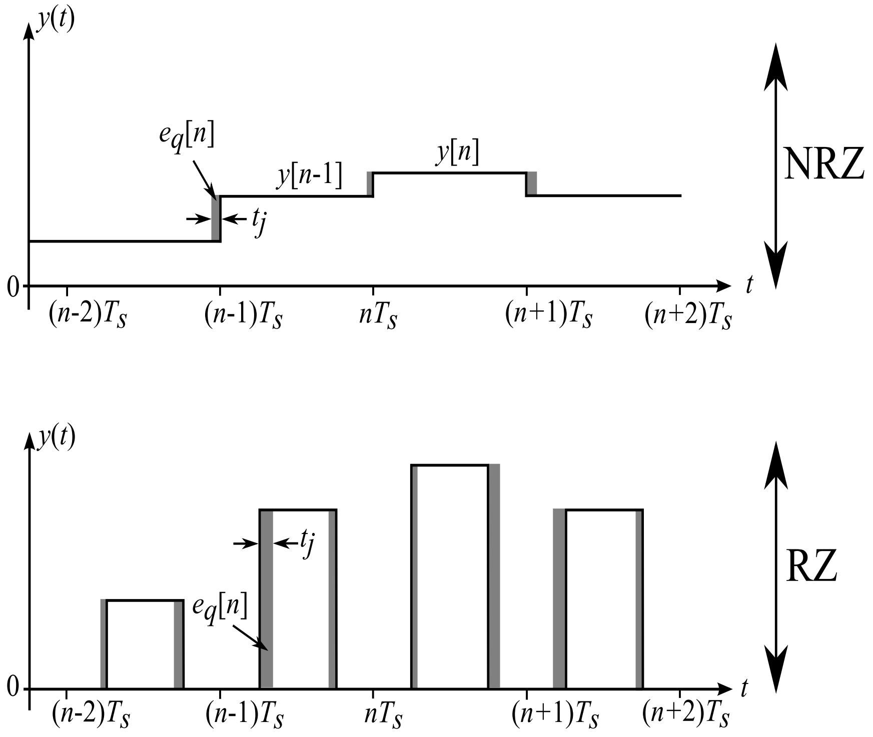


Figure 2.8: Multi bit DAC pulses affected by clock jitter

One method to increase jitter performance in a $\Delta\Sigma$ modulator is to use a multi bit internal quantizer, hence the DACs will be multilevel as well. From figure 2.8 we see that the charge error decreases drastically for NRZ pulses and less for RZ. This is due to the difference between two adjacent intervals will be only one LSB, hence the value of the level, determining the height of the rectangle area, decreases. This is advantageous for NRZ since the transitions only have to go from one level to the adjacent one. For RZ pulses the transitions always are between zero and the level instead, although there is an improvement as the average height of the rectangles decreases. It is also worth mention that $A_{f,NRZ}$ in this case will be 1 since increasing the number of bits make it less likely for non transition to happen. Still for NRZ the noise improvement is about $6dB$ for each bit, which is like the quantization improvement, while for the RZ counterpart it is only a minor improvement [21].

2.2 Loop filter realization for low power applications

In this section we will discuss how the loop filter of the modulator can be implemented, since our focus is on low power applications we will cover techniques to reduce the power consumption. The loop filter is the most power hungry part in the ADC and it is easy to understand why, as every integrator has an amplifier. So the majority of active elements are inside the loop filter, and also it's desirable to have an high loop gain and bandwidth which lead to high power consumption. Hence being the loop filter the most power hungry part of the modulator reducing its power consumption will bring the most benefits to the whole system.

2.2.1 biquad

The standard way to implement the loop filter is to cascade single integrators stages and/or biquad cells 2.9. This structure is the most suitable for low power applications [28]. We can see there are only two feedback paths, one going to the

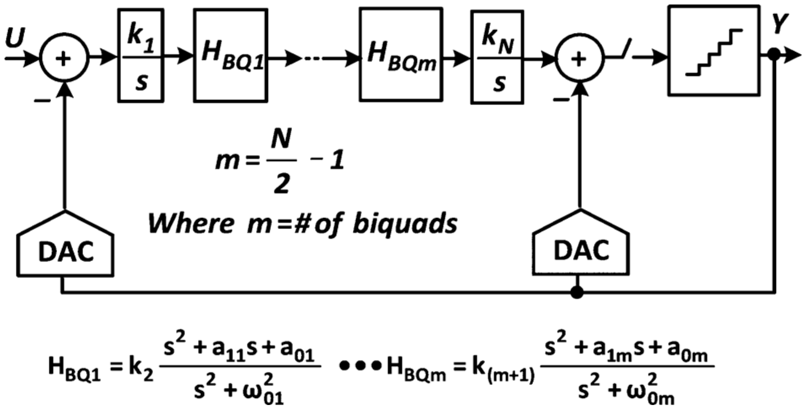

U
+
−
$\frac{k_1}{s}$
H_{BQ1}
H_{BQm}
$\frac{k_N}{s}$
+
−
Y
$m = \frac{N}{2} - 1$
Where m=#of biquads
DAC
DAC
$H_{BQ1} = k_2 \frac{s^2 + a_{11}s + a_{01}}{s^2 + \omega_{01}^2}$ ••• $H_{BQm} = k_{(m+1)} \frac{s^2 + a_{1m}s + a_{0m}}{s^2 + \omega_{0m}^2}$

Figure 2.9: N-th order modulator with cascaded biquad cells

input, global feedback, and a zero order feedback path needed to compensate for

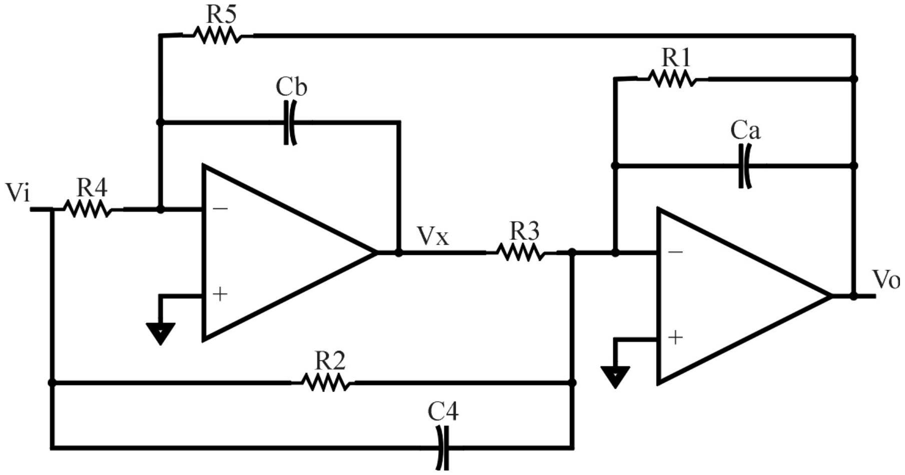


Figure 2.10: Biquadratic cell

loop delay. Also the last integrator is omitted if N in odd. The most common way to implement the biquad cell is using active RC integrators due to their high linearity hence every biquad has two amplifiers to achieve a second order transfer function, figure 2.10. To reduce the number of amplifier of the loop filter, and so the power consumption of the system, it is possible to implement the structure in figure 2.9 using active integrators and passive networks. Passive networks don't consume power but they also attenuate the signal and hence increase the noise. Then the active blocks will need to suppress the noise introduced by the passive block. The design

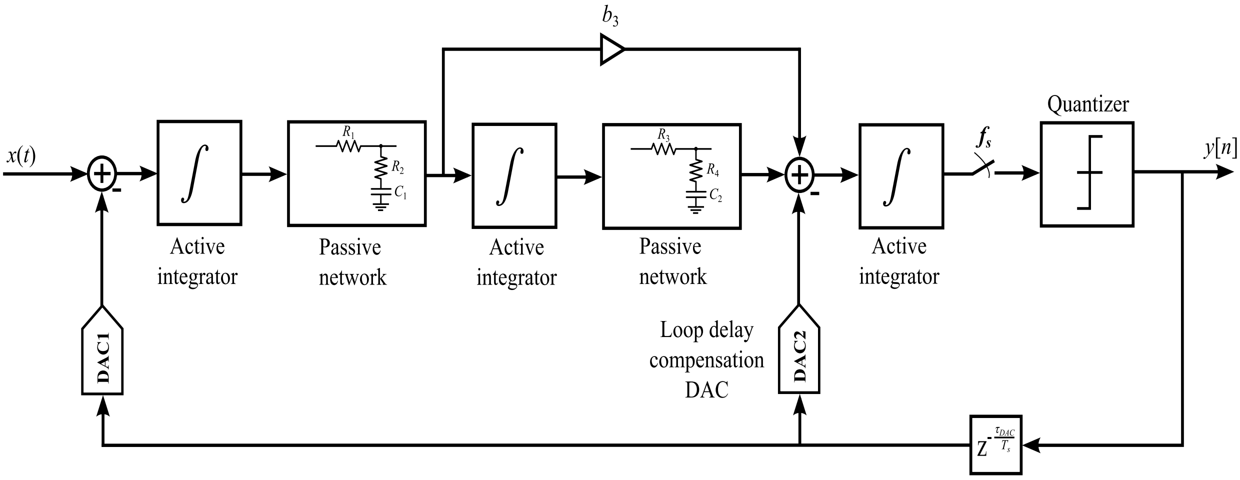


Figure 2.11: Diagram of a 5-th order modulator using active and passive blocks

of this kind of architecture is done by imposing the cut-off frequency of the passive filters and the gains of the amplifiers such that the desired NTF is obtained.

2.2.2 Single opamp networks

It is possible to implement the loop filter in a way that combines both advantages of active networks and passive networks. This was presented in [22] and it consists of a passive network and a single amplifier and it's called single opamp network (SOA). The general scheme for this network is shown in figure 2.12

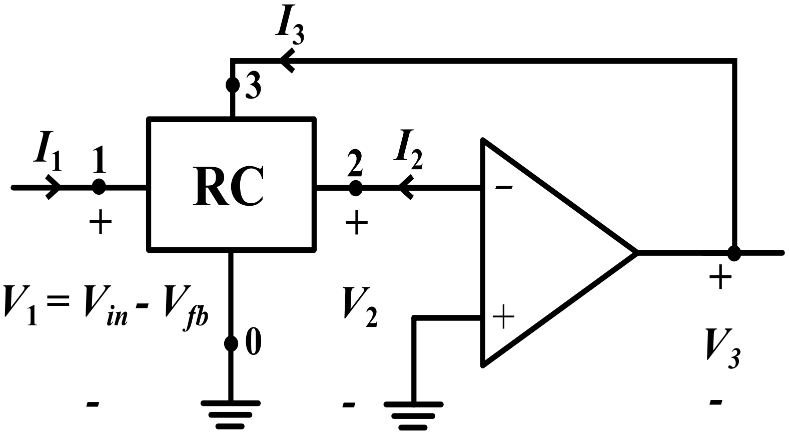


Figure 2.12: SOA representation

Here the amplifier provides the gain, hence there is no increase in the noise while the RC network can be shaped to achieve more aggressive noise shaping performance. To be able to substitute one SOA in the diagram in figure 2.9 the network must realize a transfer function of the type:

$$H_{CT}(s) = k\frac{s^2 + b_1 s + b_2}{s^2 + \omega_0^2}$$

Not having a first term in the denominator of the transfer function means there's no leakage and the poles are on the imaginary axis in other words it's a resonant transfer function, this can then be used to implement a NTF with optimized zeros.

The first SOA presented in [22], is shown in picture 2.13. The transfer function implemented by this network is:

$$H_{CT}(s) = -\frac{\frac{C_{in}}{C_2}s^2 + \frac{1}{C_2 R_{in2}}s + \frac{1}{C_1 C_2 R_1 R_{in1}}}{s^2 + \frac{1}{C_1 C_2 R_1 R_2}} \tag{2.7}$$

This network then implements a second order transfer function using only one amplifier and can be used for implementing optimized zeros NTF as it has a resonant transfer function. On the other hand it can be noted that this network is a 3rd order transfer function that is reduced to the second order TF in (2.7) when $R_3^{'}||R_{in2} = R_1||R_2||R_{in2}||R_{in1}$ and $C_3 = C_1 + C_2 + C_3$. Satisfying these two condition is difficult task due to the process variations. The outcome is that the NTF will have different coefficients and thus degrading the performance of the system and it's stability. Another thing that should be noted is that this network can't

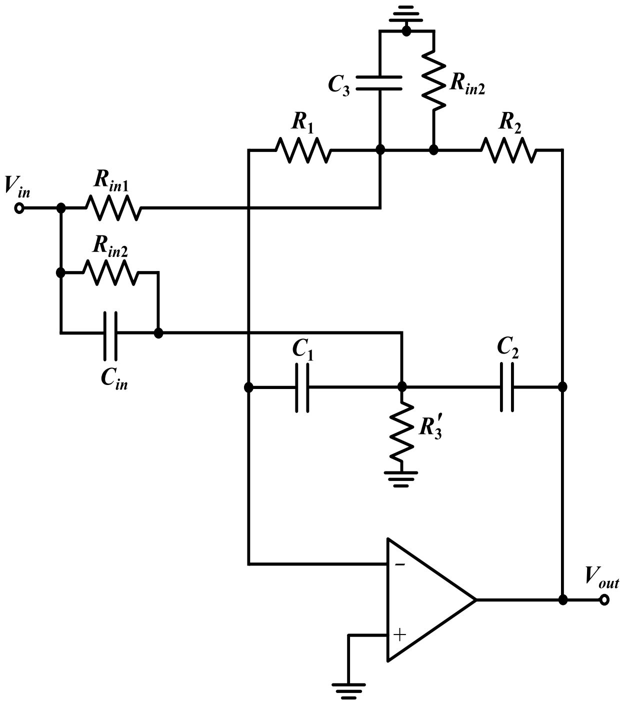


Figure 2.13: First SOA presented

be used as front end in the modulator, in fact the input signal is not applied to virtual ground hence it is not possible to subtract the feedback signal directly from the input as can be done commonly when the input is applied to the amplifier directly.

Another SOA network was proposed in [40], figure 2.14 with transfer function:

$$H_{CT}(s) = -\frac{s^2 + \frac{1}{C_1}\left(\frac{2}{R_1} + \frac{3}{2R} + \frac{1}{2R_2}\right)s + \frac{1}{C_1C_2R_1}\left(\frac{1}{R_2} + \frac{1}{R}\right)}{s^2 + \frac{2}{C_1R_3}s + \frac{1}{C_1C_2R_3}\left(\frac{1}{R_2} + \frac{1}{R}\right)}$$

This network realizes a true second order transfer function, hence the first order term in the denominator means the two poles will be moved into the left half plane from the imaginary axis. This means that this network won't be able to implement optimized zeros for the NTF. Although choosing an high value of R_3 reduces the leakage and mitigate the effect of not having poles on the imaginary axis. Compared to the first network this has less passive elements, which saves area and reduces the noise referred to the input.

A third SOA network is shown in figure 2.15 and was presented in [41]. We can already compare this with the previous ones and see it uses less components than the previous two, the input is also applied to the opamp input and hence it can be used in the first stage of a modulator, hence it has all the advantages of the second

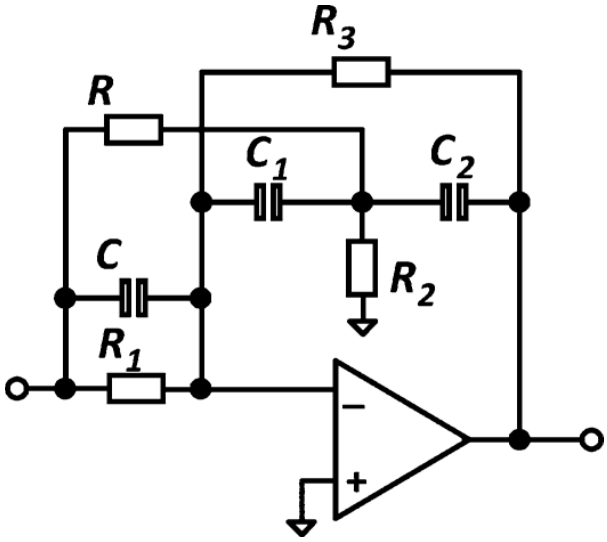


Figure 2.14: Second SOA presented

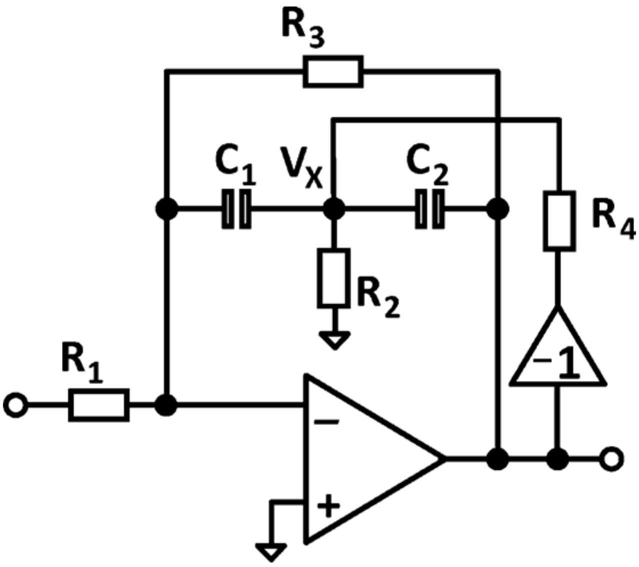


Figure 2.15: Third SOA presented

network. Its transfer function is:

$$H_{CT}(s) = -\frac{1}{C_1 C_2 R_1} \cdot \frac{(C_1 + C_2)\, s + \left(\frac{1}{R_2} + \frac{1}{R_4}\right)}{s^2 + \frac{1}{C_1 C_2 R_3}\left(\frac{1}{R_2} + \frac{1}{R_4}\right)}$$

The transfer function is a resonant one, there is no first term in the denominator and so the poles are on the imaginary axis, this holds true when:

$$R_3C_1 = R_4\left(C_1 + C_2\right). \tag{2.8}$$

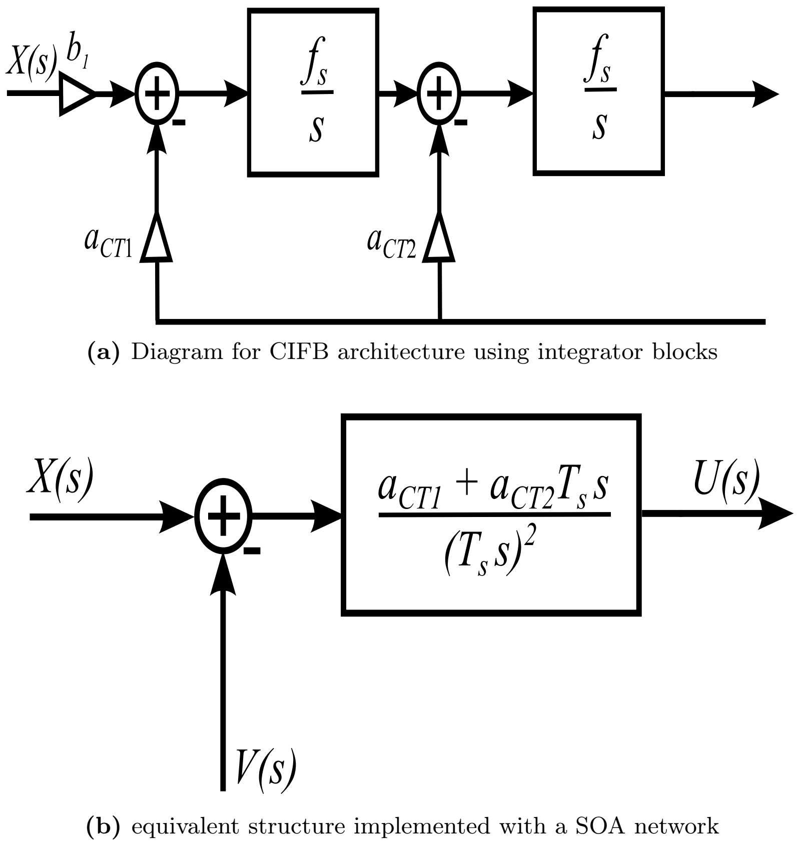
When the condition is met the feedback path through R_4 eliminates the leakage term and allow the network to be used to implement a zero optimized NTF. Opposed to the condition to have a resonant function for the first network (2.8) can be written as

$$\frac{C_1}{C_1 + C_2} = \frac{R_4}{R_3},$$

which means matching a capacitances and resistances ratio which can be done with up to 0.1% and 1% precision respectively in a CMOS process. This allow the network to implement the desired transfer function, thus the NTF, whit a good precision.

2.2.3 AA properties using SOA

We now compare the standard architecture for the loop filter with an equivalent one made using a SOA network, in figure (2.16) a second order loop filter is shown and implemented in both ways, the analysis can then be easily extended to higher orders.



(a) Diagram for CIFB architecture using integrator blocks

(b) equivalent structure implemented with a SOA network

Figure 2.16

For these systems two transfer functions exist, one from the input to the output and the other from the feedback to the output. In both cases the feedback transfer function is:

$$H_{CT}(s) = \left.\frac{U(s)}{V(s)}\right|_{X(s)=0} = -\frac{a_{CT1} + a_{CT2}T_s s}{(T_s s)^2}.$$

In fact the two systems are equivalent, they have the same NTF and hence the feedback transfer function is the same. On the other hand it is easy to see that for the feedforward transfer function this is not true, in the the case of a SOA structure the feedforward function is equal to the feedback one, aside from a minus sign:

$$G_{CT}(s) = -H_{CT}(s),$$

while for the CIFB structure it is

$$G_{CT}(s) = \frac{b_1}{(T_s s)^2}.$$

Hence G_{CT} in the CIFB case has no zeros while it has one zero in the SOA case, thus the $G_{CT}(s)$ will decrease more slowly in the SOA case. As it was shown

$$STF(s) = G_{CT}(s)NTF(e^{sT_s})$$

hence also the STF will reduce more slowly when using a SOA network. The implication of this behavior is a less AA effect for the SOA network. In figure 2.17 a zooming of the first aliasing zone is shown, we can see that in the SOA the STF is always higher in value than the CIFB one, hence degrading the AA performance of the system. Also the STF for a SOA network will depend on the coefficients a_{CT1} and a_{CT2}, hence when applying the loop delay compensation, changing these coefficients

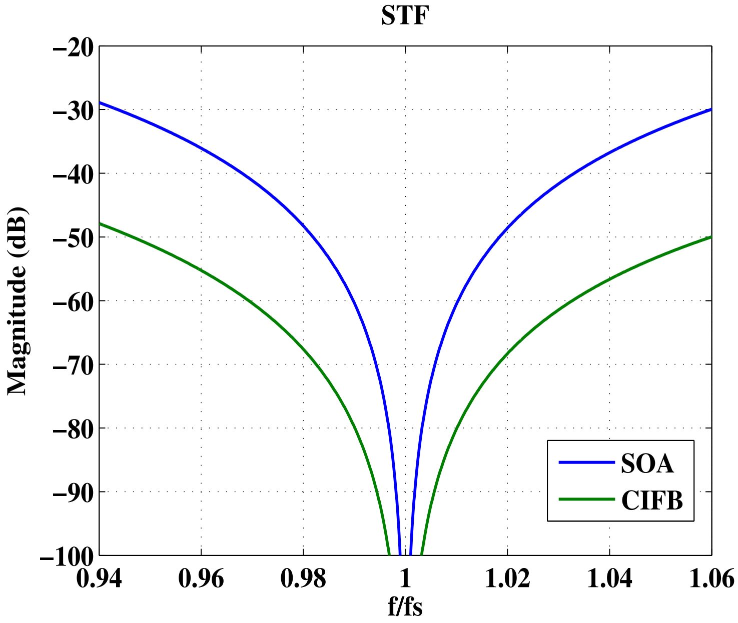


Figure 2.17: Zooming of the first aliasing zone with the STFs for the two architectures

the behavior of the STF is changed as well. The STF can then be different from unity in band and also present a peaking (the non unity gain in band also arises only in the case of RZ DACs), in the next chapter we will see how the third SOA network can be used to restore partially control of the STF.

2.3 Internal quantizer

In this section we will see the working principles of a quantizer based on a successive approximation register (SAR quantizer). This implementation will be used in this project as it allow to use only one comparator to implement a multibit internal quantizer. Why this is convenient compared to a flash architecture is also discussed briefly here, hence the flash architecture is also discussed briefly now.

2.3.1 Flash quantizers

Flash quantizer are the fastest quantizers, a typical implementation is shown in figure 2.18, their working principle is to divide the input range in $2^b - 1$, where b is the number of bits levels then compare the input signal with every level, hence determining which level is the best approximation of the signal. The number of the level which is closest to the input is then coded into binary code using logic circuitry. This is the fastest architecture for a quantizer since the comparisons with each level and the input are done at the same time, on the other hand to carry all the comparisons at the same time requires one comparator for each level, thus $2^b - 1$ comparators. Flash quantizer was used in previous implementations of the modulator, [30, 31], in this version to further improve the power consumption a SAR quantizer was used, a few $\Delta\Sigma$ modulators were implemented using an internal SAR quantizer, in [19] this was done for audio applications for a DT modulator and in [32, 36] it was used in a CT architecture as it will be seen in next section in CT implementations a SAR quantizer will introduce a loop delay far greater than a flash architecture as it requires more clock cycles for its operation.

2.3.2 SAR quantizers

In a flash architecture the $2^b - 1$ comparisons are made all at the same time using more comparators, in a SAR quantizer the approach is different. Instead of using more comparators only one comparator is used to compare the input with a desired level that is generated by a DAC and will change every cycle. The level that best approximate the input is then searched varying the comparator input. The research is like a binary search algorithm, the first step determine if $V_{in} \geq \frac{V_{fs}}{2}$, where V_{in} is the input signal and V_{fs} if the full scale voltage of the quantizer. The next step depends on the comparison ran in the first one, if $V_{in} \geq \frac{V_{fs}}{2}$ then the input signal will be compared with $\frac{3}{4}V_{fs}$, else with $\frac{1}{4}V_{fs}$. The conversion will then continue for b steps to achieve a b bit resolution. In fact the first comparison will yield the MSB bit value and the others the remaining bits till the LSB during the last one comparison. A scheme of a SAR quantizer is shown in figure 3.15, the input is compared with a voltage coming from a DAC which is controlled by a control logic. The SAR logic keeps track of the actual cycle of the conversion and the output of the previous

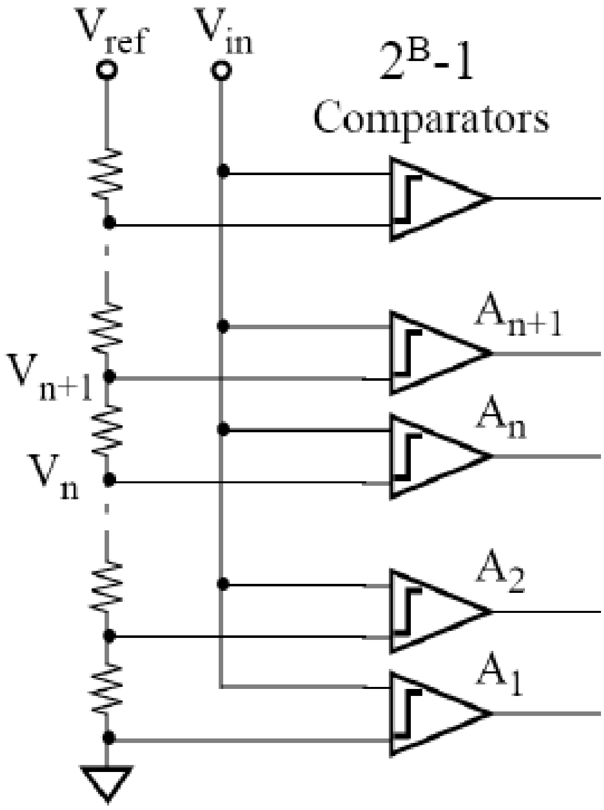


Figure 2.18: Flash architecture with resistive ladder and $2^b - 1$ comparators

comparisons and update the DAC output accordingly. In this process the input has to be stable, hence a sample and hold block is also needed at the comparator input. A representation of the comparator inputs during each cycle is shown in figure 2.20 With this architecture it is possible to achieve a desired resolution using only one comparator, the drawback is that the conversion takes several clock cycles: $b + 1$, one conversion for every bit and also one more cycle to sample the input. The SAR will have to be clocked faster than the sampling frequency of the whole modulator then and this can greatly increase its power consumption, as the comparator and the logic will have to be fast. The solution we adopted is to use an asynchronous quantizer, hence the comparator clock will be generated by the SAR logic, this avoid the need for another clock, still the major limitation in increasing the resolution is the speed of the logic and the loop delay that will result from it, thus the number of bit can't be too high, we used for this reason a 4bit quantizer.

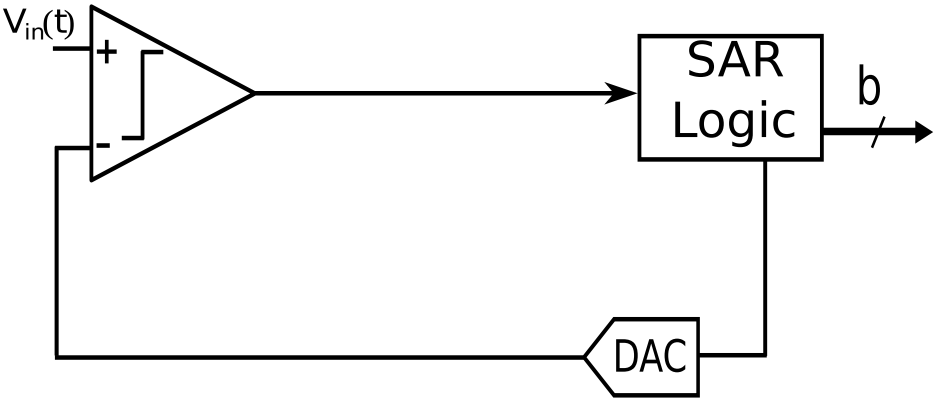


Figure 2.19: SAR quantizer architecture

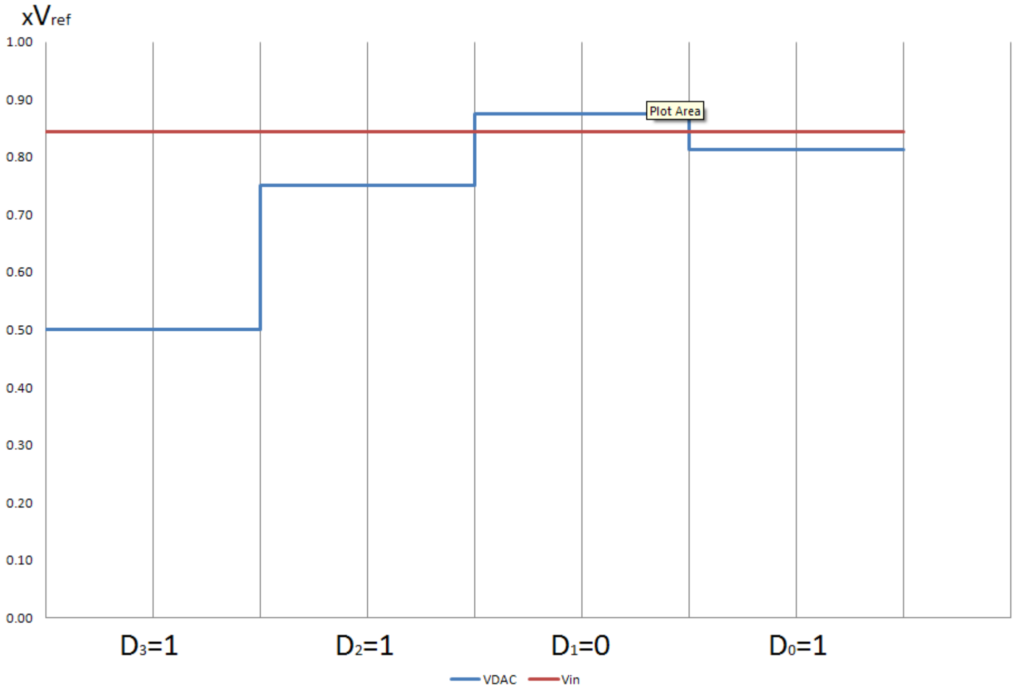


Figure 2.20: SAR quantizer comparator input signals during its operation

Chapter 3

Proposed modulator

After reviewing the principles of the $\Delta\Sigma$ modulation the project carried out for this thesis is presented in this chapter.

3.1 The UDP project

This project has been part of the ultra-portable devices (UDP) project founded by the Swedish Foundation for Strategic Research, the project aims to develop a receiver for short range wireless communication in ultra low power devices. The main specifications of the receiver are:

- Power consumption: $1mW$
- Data rate: $250kb/s$
- Frequency band: $2.45GHz$
- chip area: $1mm^2$

The receiver chain, figure 3.1 consists in a front end which amplifies the small radio signal using a LNA amplifier, then the signal is brought to baseband using a mixer. All of this takes place in the analog world, the signal is then converted to a digital signal and the information it carries are elaborated. To convert signals

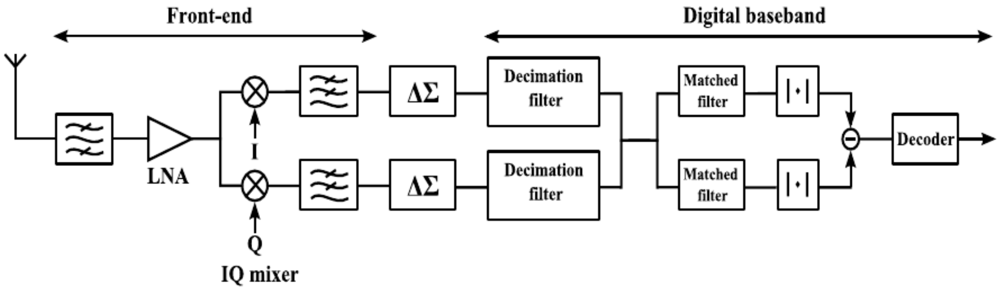


Figure 3.1: Full diagram of the UDP receiver

from analog to digital the ADC is needed and a mandatory part of a receiver, this is the part designed here, the project was first took by a former PhD student at Lund University(Sweden), Dejan Radjen, who made different versions of the converter[[29]]. The last one of those fabricated had some stability issue and was not possible to measure its performance. The focus of the thesis is to redesign the loop filter of the modulator and finding and solving the problem that prevented the circuit to be measured.

This work is then a follow up of this PhD dissertation, hence the specifications at system level for the ADC were already determined through simulations at system level of the receiver. The ADC block will have to fulfill the following requirements:

- Power consumption $\leq 100\mu W$
- $SNDR \geq 60dB$
- Signal Bandwidth $f_b = 500kHz$
- $V_{DD} = 800mV$

The architecture for the ADC block was chosen to be a CT $\Delta\Sigma$ due to its low power capabilities discussed in the previous chapter.

3.2 Specifications of the ADC

From the requirements different solutions to meet them were explored, various configurations for a $\Delta\Sigma$ modulator were simulated using matlab and Schreier's toolbox [33]. Reducing the power consumption is the main aim of the project and the choices made focuses on lowering it as much as possible. Hence the OSR was kept as low as possible for reducing the sampling frequency which affects the power consumption of the system since faster circuits are needed for high OSR, and this increases the power consumption drastically, to achieve high OSR in fact would require to lower the signal bandwidth to keep the sampling frequency low enough. It has been showed though [11] that a minimum OSR of about 4 is required to benefit from the advantages of the $\Delta\Sigma$ modulation.

Regarding the order of the modulator, higher orders allow to realize more aggressive NTF and then increasing the noise attenuation in band compared to lower orders, on the other hand a high order will require an high number of amplifiers as well, which as said are the most power hungry part of the system. Thus a trade off must be found on the order of the modulator, and the performance of the NTF. In this project the SNR requirement is quite relaxed so there is no need for a high order modulator.

The quantizer will also affect the power consumption and the overall performances of the system, its choice will affect greatly the whole quantizer then. The quantizer is chosen to be a multibit one, this allow to use a lower OSR and lower order NTF as its quantization noise is lowered compared to a 1bit quantizer by about $6dB$ for every every bit. Also a multibit quantizer results then in a modulator with a better stability as it will have a greater stability range. The architecture of a multibit quantizer can though greatly affect the power consumption, for instance a B bit flash quantizer will require $2^B - 1$ comparators. Each comparator will increase the power consumption and their number increases exponentially with the number of bits, thus limiting the number of bits when power consumption is of a major concern. Also the high number of comparators will load the output of the loop filter with a capacitive load given by the input stage of the comparators, this load increases exponentially with B then. This capacitive load affects the power consumption of the loop filter, that will be required to give enough current to drive it. The chosen architecture for the quantizer was then a successive approximation register (SAR), this allows to use only one comparator but that need to be clocked with higher speed compared to the sampling frequency. The SAR principles and its implementation will be discussed later in this chapter.

A suitable configuration found from the simulations using the matlab toolbox consists in an internal quantizer of 4 bits, a second order modulator and $OSR = 16$. The benefits of this configuration are quite straightforward. The OSR will result in $f_s = 16MHz$ which for the $65nm$ CMOS technology that will be used for this project is reasonable for low power application. A second order loop filter is a great advantage as it was implemented using SOA, this leads to having only one amplifier for the whole circuit. A spectrum of the output of the simulated system is shown in figure 3.2 As can be seen the ideal $\Delta\Sigma$ modulator achieve a maximum SNR of

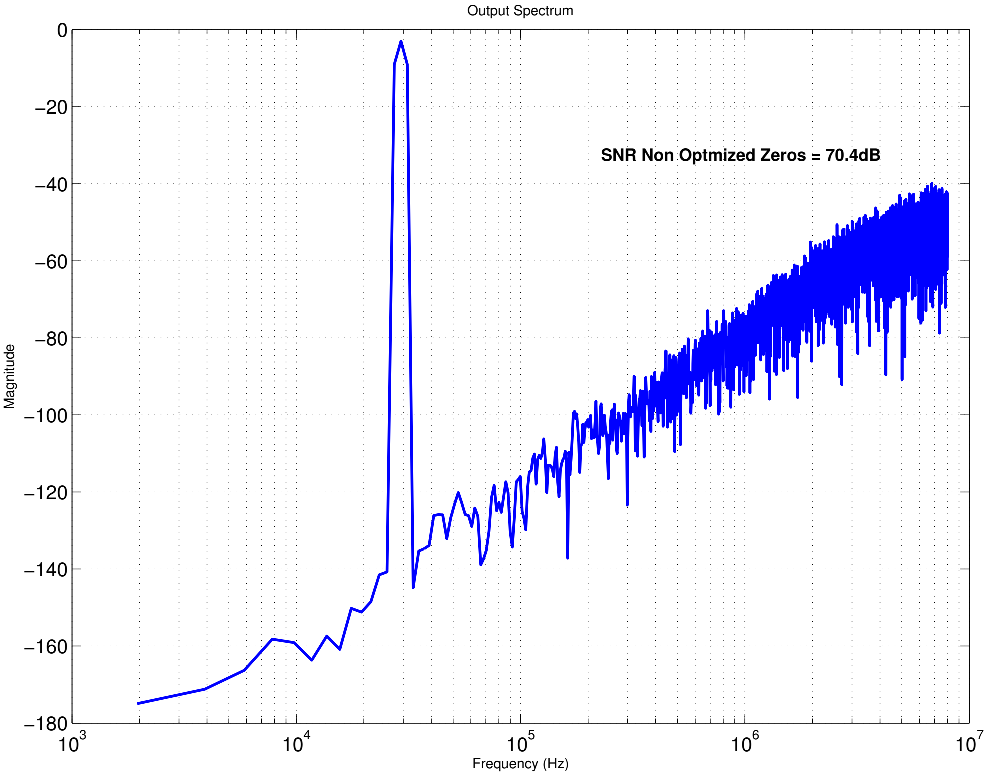


Figure 3.2: Ideal $2nd$ Order $\Delta\Sigma$ modulator with $4bit$ quantizer, $f_s = 16MHz$ and $f_b = 500kHz$

$70dB$, this was intended as the requirement specifies $SNR \geq 60dB$ and $10dB$ are kept as a margin for non idealities such as thermal noise and jitter effects. It should be noted from the shape of the noise that the NTF doesn't have optimized zeros, in figure 3.3, the different shape of the noise due to the optimized zeros can be seen and also that having optimized zeros brings an increase in SNR of about $3dB$ since the noise is better suppressed all across the signal bandwidth.

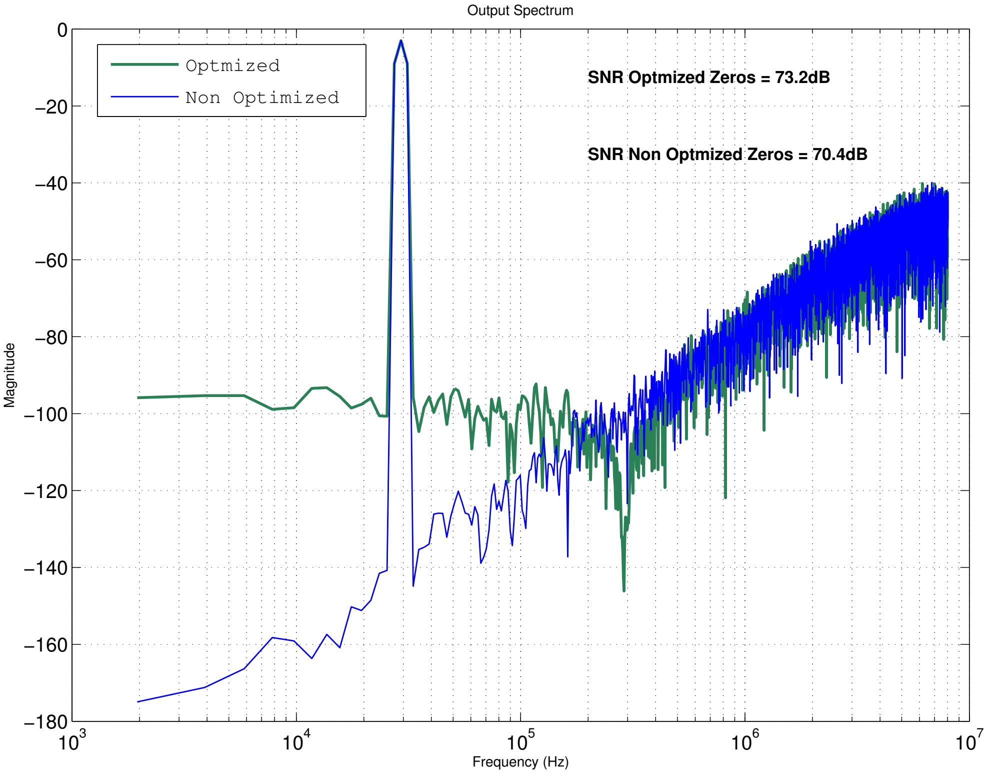


Figure 3.3: optimized ntf vs non optimized

In the next sections the choices made to implement this modulator will be presented and explained.

3.3 Architecture

The architecture chosen to implement the loop filter of the modulator is reported in this chapter also, in figure 3.5, this structure is the most suitable for low power applications, [28].

To calculate the a_{ct1} a_{ct2} coefficients for our system we start from the discrete time NTF, which for our system is the classical 2nd order noise shaping transfer function:

$$NTF(z) = (1 - z^{-1})^2. \tag{3.1}$$

In fact our modulator doesn't implement optimized zeros and hence the NTF has the zeros on the unity circle. From the discrete time NTF to be able to obtain the

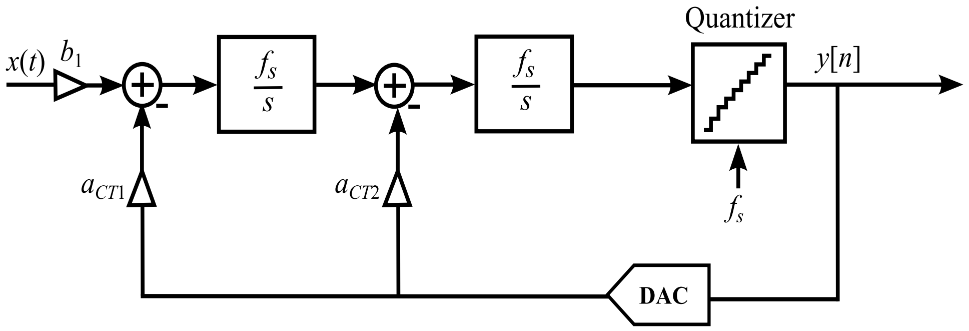


Figure 3.4: CIFB structure of a second order modulator

continuous time coefficients we apply the impulse invariant transformation, to do so we must know the DAC impulse response. In this work a NRZ DAC was chosen to implement the global feedback path, the reason for it is to reduce the effect of clock jitter, in fact in the modulator there will be only the first feedback path as we use a SOA network to implement the loop filter. As it was shown in the previous chapter the NRZ DAC will be more affected by the loop delay since its pulses are always pushed into the next sampling period but this can be solved by imposing a fixed loop delay as will be done later. From the NTF in (3.1) and a NRZ DAC pulse, using the method proposed in [21] the coefficients for the CT architecture are found to be: $a_{ct1} = 1$ and $a_{ct2} = 2$. Taking into account also the loop delay the structure of figure 3.4 becomes the one in figure 3.5.

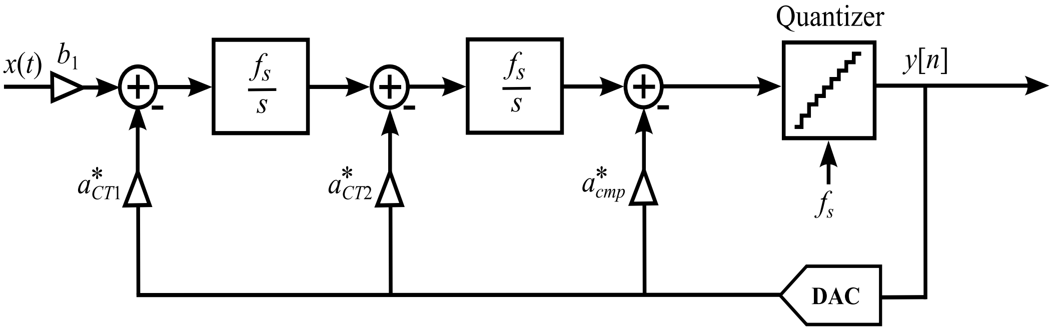


Figure 3.5: CIFB structure of a second order modulator with loop delay compensation path

Now equations 2.2 can be used to get the values in presence of loop delay, there are two ways for it:

- One is to measure the worst case loop delay and find the coefficients then. In this case the DAC pulse will never go into the next sampling period since the worst case is used. This method require to know already how the modulator will be implemented.
- A second methodology is to impose a loop delay, for example using clocked flip flops, this has to be then large enough to ensure proper working also in worst

case scenario.

The latter approach is used ad it allows to vary the loop delay to adjust a^*_{cmp} to a desired value, between the time boundaries imposed by the circuit to ensure proper working. This is desirable as the compensation coefficient is implemented in the path that goes from the output of the quantizer and its input. At the quantizer input it is also desirable to avoid an active summer block that will consume power, hence an easy way to implement the summing operation is using two capacitive DACs. In the modulator design a SAR quantizer is used for its low power capabilities, and hence a capacitors array is used, which makes it straightforward to add a capacitive DAC to sum the feedback signal. Now to ease the design process we can set $a^*_{cmp} = 1$ which will result in two identical capacitive DACs. Imposing $a^*_{cmp} = 1$ and using the coefficients previously found lead to the new coefficients:

$$\begin{cases} a^*_{CT1} = 1 \\ a^*_{CT2} = 2.06 \\ \tau_d = 0.56 \end{cases}$$

where τ_d is again normalized loop delay. The loop delay results then in $T_d = 35ns$, this is more than half clock cycle and seems reasonable to ensure proper operation of the quantizer and the DAC. Simulations at the end shows that this is the case.

3.4 Loop Filter

After determining the right coefficients for the architecture we will show how they are implemented in the loop filter. As was discussed in 2.2 the loop filter was implemented using SOA networks to reduce the number of amplifiers and through it the total power consumption of the system. The chosen network was the third presented in chapter 2 and shown in figure 3.6. Being a second order modulator only one SOA network is required, in fact its transfer function is second order with resonant poles, its equation is recalled here:

$$H_{CT}(s) = -\frac{1}{C_1C_2R_1} \cdot \frac{(C_1 + C_2)\, s + \left(\frac{1}{R_2} + \frac{1}{R_4}\right)}{s^2 + \frac{1}{C_1C_2R_3}\left(\frac{1}{R_2} + \frac{1}{R_4}\right)}.$$

Hence the network will be used in the front end of the modulator, indeed only one network is used, the thermal noise of this stage is the most relevant one and to reduce its power a different version of the network is used, removing the resistances R_3 and R_4 the network will still implement a second order transfer function, but with reduced thermal noise power at the input. Reducing the thermal noise in this stage is critical to keep it low enough to achieve the desired performance, in fact the thermal noise here is not shaped by the modulator because it is added directly on the input signal. The resulting network is shown in figure 3.7, where also the global feedback path from the DAC is shown, and also a common value R is used to simplify the design and three coefficients n_1,n_2,n_3 will determine the actual value of each resistor according to the a^*_{ct1} a^*_{ct2} a^*_{cmp} coefficients and hence the ones of the NTF.

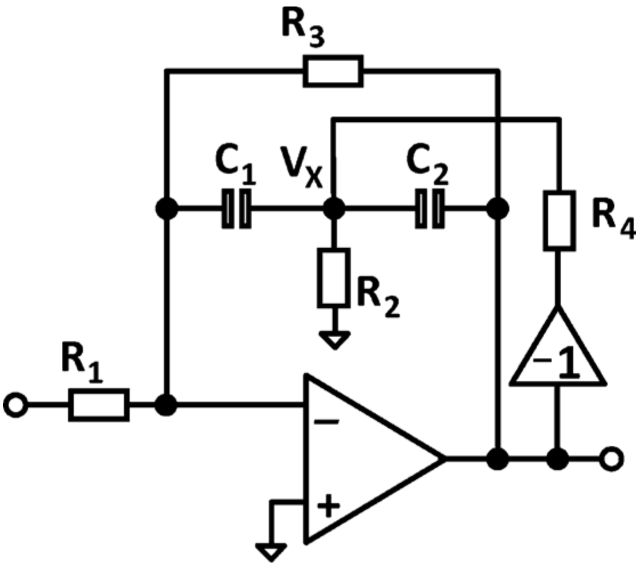


Figure 3.6: Third SOA presented

Since this network implement a resonant couple of poles through feedback path with R_4, removing this path will result in having a non optimized zeros in the NTF, meaning that it's not possible to move the poles from the origin along the imaginary axis. This is not a problem in the design as using non optimized zeros we can still achieve a SNR of $70dB$ which leave enough room for non idealities in the ADC.

In this network we can identify has was shown in section 1.2 for the general case, two transfer functions: feedforward and feedback transfer functions. These two will affect the STF and the NTF of the modulator, and as it was shown in section 2.2.3 in case of a SOA network the AA property of the CT modulator is degraded since the STF will have one zero. The two transfer functions are given by:

$$V_{out} = -\left(\frac{\frac{1}{n_1 n_3} + \frac{2}{n_3} sRC}{(sRC)^2} V_{in} - \frac{\frac{1}{n_1 n_2} + \frac{2}{n_2} sRC}{(sRC)^2} V_{fb}\right)$$

We can see that coefficient n_2 will affect only the feedback transfer function, hence it will be determined by the NTF, the coefficient n_3 on the other hand is only present in the feedforward transfer function, hence it can be changed to control the STF behavior without affecting the NTF. This will help to reduce the degradation of the AA properties, even though it won't fully restore them. In figure 3.8 we can see how using the n_3 coefficient the STF can be controlled and the peaking at the band transition reduced when compared to a SOA network without STF controlling coefficient. To finally find the values for the network the feedback transfer function of the network is matched with

$$H_{CT}(s) = -\frac{a^*_{CT1} + a^*_{CT2} T_s s}{(T_s s)^2}$$

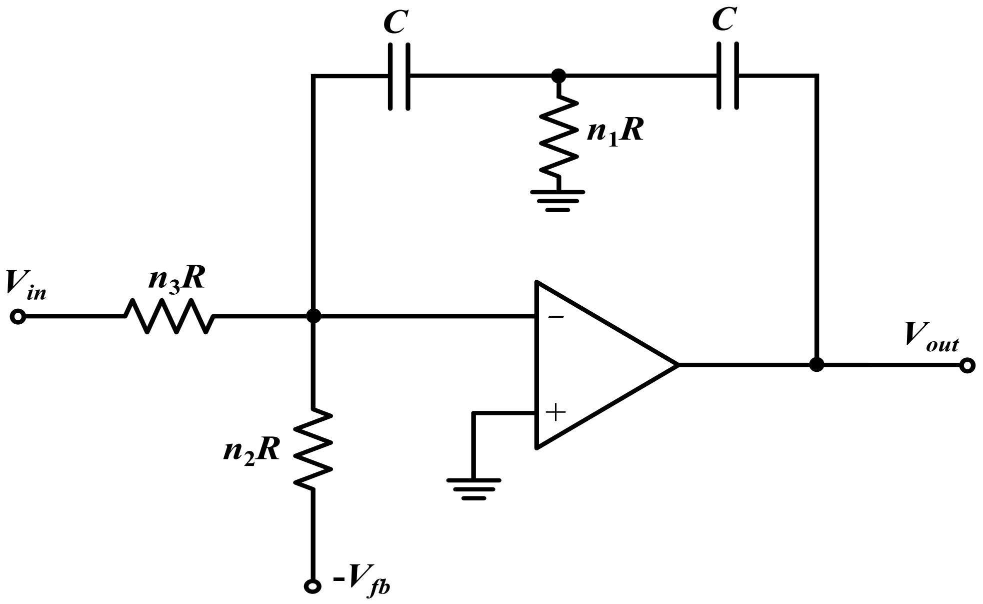


Figure 3.7: Implemented SOA network with signal input and feedback paths

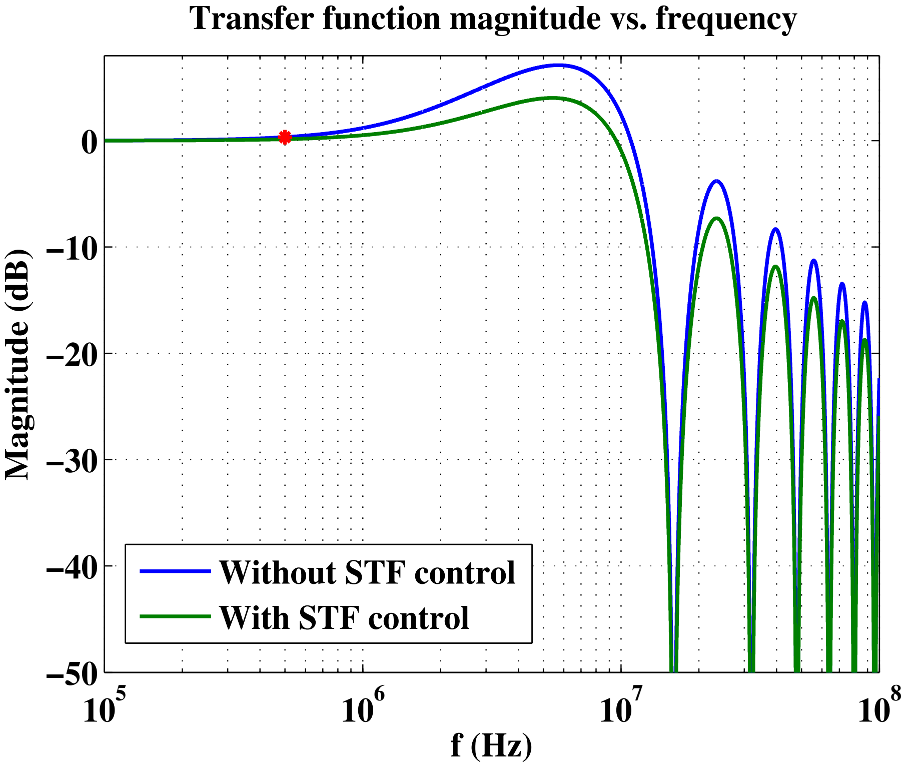


Figure 3.8: STF magnitude plot for a SOA network with STF control and one without

thus yielding

$$\begin{cases} n_2 &= \frac{2}{a^*_{CT2}} \\ n_1 &= \frac{a^*_{CT2}}{2a^*_{CT1}}. \end{cases}$$

To find the value for n_3 we equal the feedforward function of the network with

$$G_{CT}(s) = \frac{b_1}{(T_s s)^2}$$

using $b_1 = 1$, which leads to $|STF(s)| = 1$, we obtain

$$\begin{cases} n_3 = \frac{1}{n_1} = \frac{2a^*_{CT1}}{a^*_{CT2}}. \end{cases}$$

The common value of R was then found through simulations, a trade off was found between the amplifier gain, thermal noise, and power consumption and area of the DAC. In fact the lower the value of R the higher the gain of the amplifier for compensating the reduced loop filter gain due to the heavy loading effect on the amplifier. On the other hand an higher value of R increases the thermal noise and also the area of the DAC, as its resistances will be bigger. There's also a beneficial effect on the DAC power consumption of the DAC because increasing its resistance will reduce the current and hence power consumption. The value of R is then chosen through simulations and will be discussed in 3.5.1 where the requirements for the amplifier taking into account the effects of R are found.

3.4.1 Frequency response of the network and finite gain effects

The simulations of the network are presented in this section, what is relevant to notice is how the poles positions is affected by the gain of the amplifier, in fact poles shifting results in NTF zeros shifting which lead to a discrepancy between the ideal model behavior and the implemented network. In figure 4.10 the position of the network poles is shown for different gain of the amplifier, we can see that for low gain values the poles are far from the origin, while increasing the gain brings the pole closer to the origin. Hence to have the desired NTF, which affects the performance and stability of the system, the gain of the amplifier has to be high enough to also bring the poles in the origin.

3.5 Design of the amplifier

For the SOA network the amplifier was designed at transistor level, then the layout of the amplifier and the network was carried out. The opamp structure was chosen to be a two stage amplifier, this is due to the low supply voltage that has to be used in the receiver, $V_{dd} = 800mV$, hence there won't be enough voltage room to use cascode structures, like folded or telescopic OTA. A two stage amplifier needs a compensation for it to be stable, Miller's compensation is the classical way of compensating such amplifiers, this requires a capacitor added between the first and the second stage and since current will be wasted to charge this capacitor this solution is not inherently good for a power saving application. As was done in [27] we will add a feedforward path to the amplifier, the feedforward will provide a zero

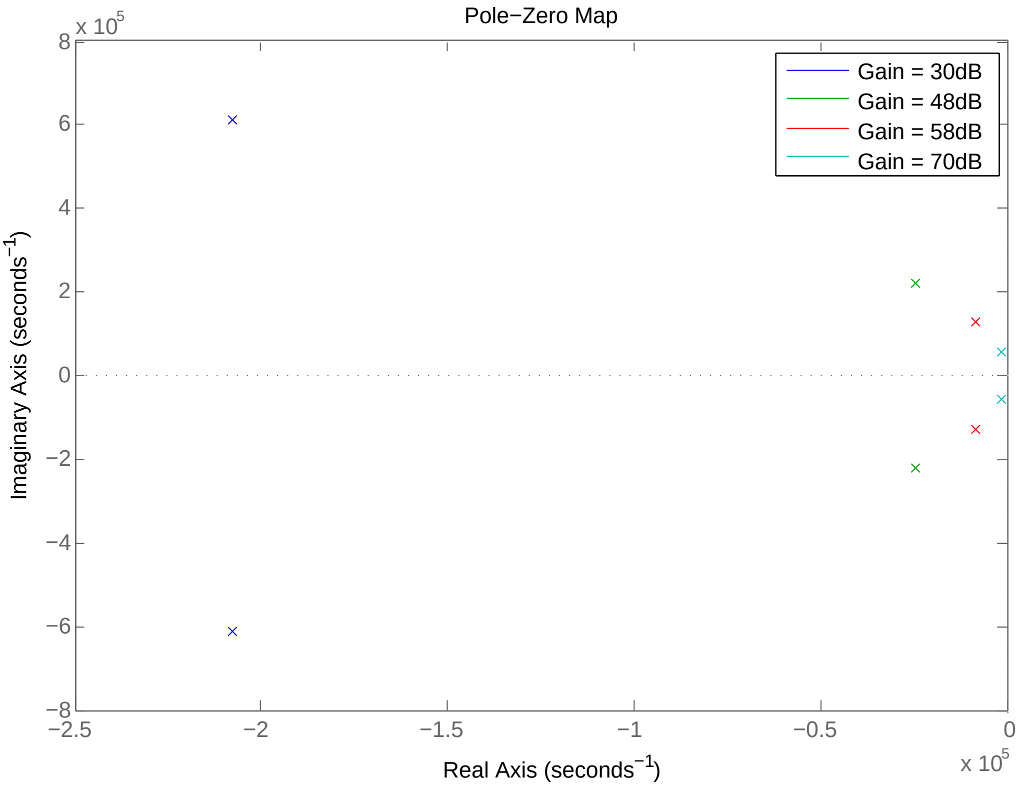


Figure 3.9: Finite gain effects on the pole of the SOA network

that can be placed such as to cancel the effects of the high frequency pole in the amplifier, hence leading to a stable system. In figure 3.10 the blocks scheme of the amplifier is shown. The schematic of the circuit that implements the structure in figure 3.10 is shown in figure 3.11. The first stage is as usual a 5mos OTA, with a pmos differential pair input stage, the output of this stage goes at the gate of a common source nmos stage, polarized using a pmos. Here the pmos of the second stage also act as the transconductance that implements the feedforward path. This doesn't increase the circuit complexity as the only added elements are R_b and C_b used to decouple the biasing voltage V_{bp} from the DC component of the input. Thus the biasing current of the second stage is reused to stabilize the amplifier. What was observed through simulations is that this compensation works well when the current of the second stage is enough greater than the one of the first stage. Also the ac coupling for the feedforward path increases the swing the maximum output swing compared to standard feedforward design compensation. The biasing is made through a current mirror, that is biased from an outside voltage, this is possible since in the chip there will be only one amplifier hence the current reference can be given as a gate voltage of a single mosfet in diode configuration. Having only one amplifier removes the need of making extra circuitry for mirroring the right biasing to different points in the circuit.

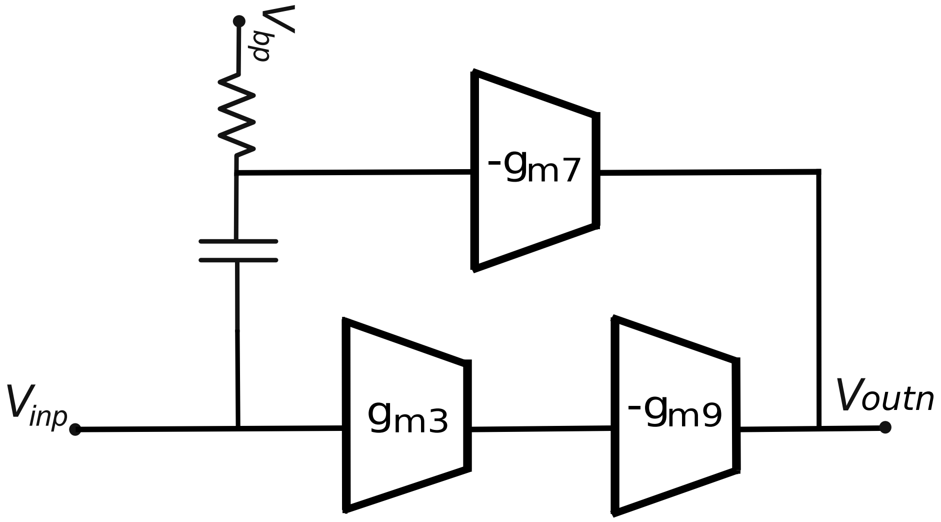


Figure 3.10: Block scheme of the amplifier

The CMFB circuity is realized using a resistive sensing through R_{cms}, high resistances values of $100k\Omega$ were used in order not to load the amplifier, this sensing as the advantage that being passive it doesn't require a biasing current and with high resistances it consumes little power as well. The common mode signal goes then into the error amplifier which is a single ended 5 mos OTA. The more classic way of implementing the actuation of the CMFB is to add a mosfet in parallel to the tail generator at the firs stage, or two mosfet in parallel with the nmos load if the input stage is made with pmos, this is to drive about a 20% of the tail current and set the correct operating point at the output. If we used two nmos in parallel with the load in the first stage (M_4 and M_5) another branch for biasing them would be required, as the bias is given by a pmos in diode configuration, the extra branch would draw current from the power supply that will increase the biasing current of the amplifier. To avoid this the load of the first stage is directly biased from the error amplifier of the CMFB circuit. The final dimensions of the opamp mosfets are reported in the table below.

	W (μm)	L (μm)
M_1	8.44	0.6
M_2,M_3	12	0.6
M_4,M_5	0.7	1.2
M_6,M_7	42	0.6
M_8,M_9	9.6	1.2

As stated early the feedforward compensation achieves good results when the current in the second stage is greater than the one in the first stage below, figure 3.12, the poles of the amplifier and the zero are shown for different values of

$$N = \frac{\left(\frac{W}{L}\right)_{M_6}}{\left(\frac{W}{L}\right)_{M_1}} = \frac{\left(\frac{W}{L}\right)_{M_7}}{\left(\frac{W}{L}\right)_{M_1}}$$

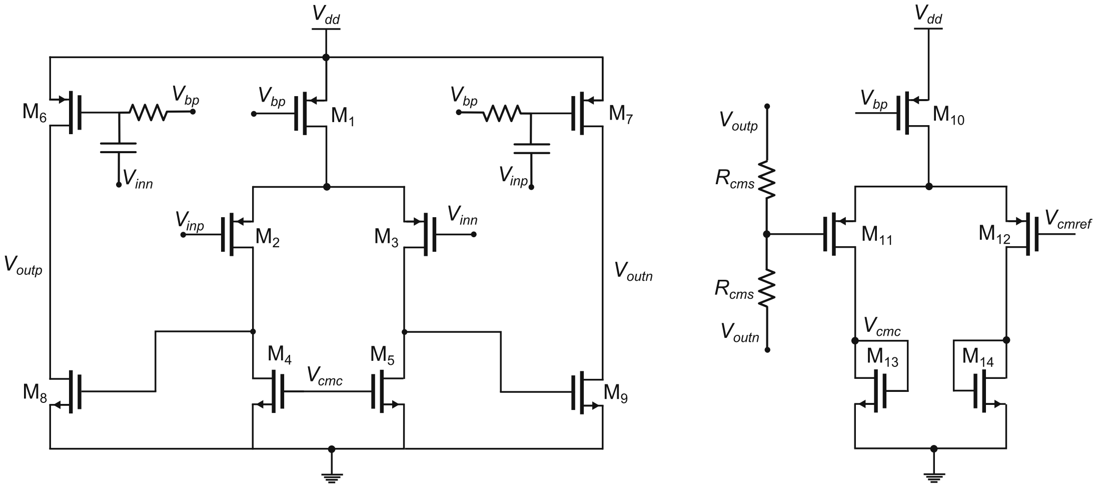


Figure 3.11: Schematic of the amplifier with the CMFB circuit

which is then the ratio between the dimension of the tail mosfet of the first stage and the ones in the second stage, thus it gives the ratio between the currents in the two stages. We see that the zero and the second pole of the opamp are quite far when $N = 1$ and the zero can't compensate since the phase effects of the pole happens too early compared to where the zero is. For $N \geq 5$ zero and pole start to get closer and hence the zero can actually compensate the pole effects on the amplifier.

As it will be shown in chapter 4 the amplifier has a gain of $66dB$ and a unity frequency $f_u = 216MHz$ while consuming only $52\mu A$.

3.5.1 Amplifier requirements

The requirements are found through simulating the loop filter with an ideal amplifier and varying its gain and GBW, this also allow to explorer the different values of resistances that can be used. The simulation included then the thermal noise contribution of the network while the model for the amplifier allowed to modify gain and bandwidth to find the minimum values to have the desired performance of the circuit. In figure 3.13 the graph shows how the SNR is affected by different choices of R and gain of the amplifier, the input signal has an amplitude $A_{in} = -3dBFS = 100mV$ which result in the peak SNR for this circuit. To ensure a maximum SNR of more than $65dB$ the gain of the amplifier has to be greater than 100 for $R \geq 15k\Omega$. We can see that increasing the value of R is beneficial as the loop gain is increased and hence there is more noise suppression, while increasing R more than $40k\Omega$ is not feasible to implement the DAC and also the thermal noise will increase and degrade the performance. A choice of $R = 30k\Omega$ is made in this circuit, this result in a reasonable DAC area and power consumption and also leads to an $SNR \geq 65dB$ for an amplifier gain of $40dB$. Furthermore increasing the gain to $46dB$ the SNR will almost reach the limit of $70dB$ with $R = 30k\Omega$. In figure 3.14 the varying of the SNR with the bandwidth of the amplifier is shown for $R = 30k\Omega$,figure 3.14, the gain is fixed and high such that it doesn't degrade the SNR and in this case to achieve an SNR close to the limit of $70dB$ a $GBW \geq 100MHz$ is needed.

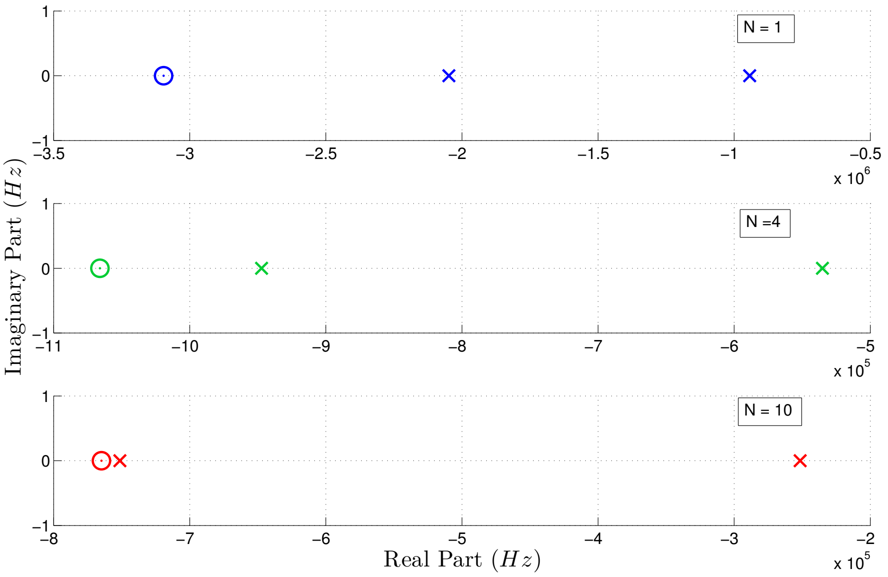


Figure 3.12: Varying of the poles and zeros in the opamp for different ratio of currents between the 2 stages

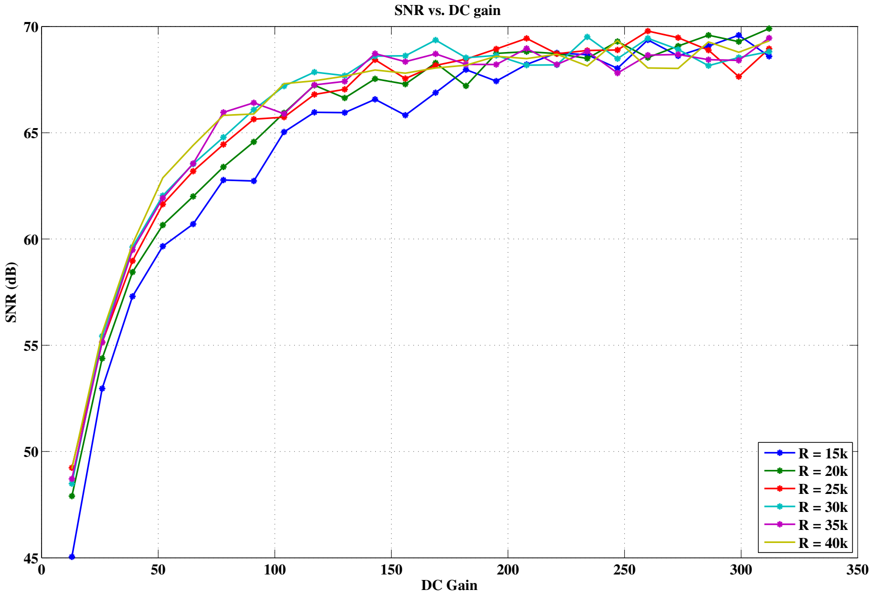


Figure 3.13: SNR for different gain values and R values

3.6 SAR quantizer

The working principles of the SAR quantizer were shown in 2.3.2, here the actual implementation used is discussed. As was said the internal quantizer is a 4 bit quantizer.

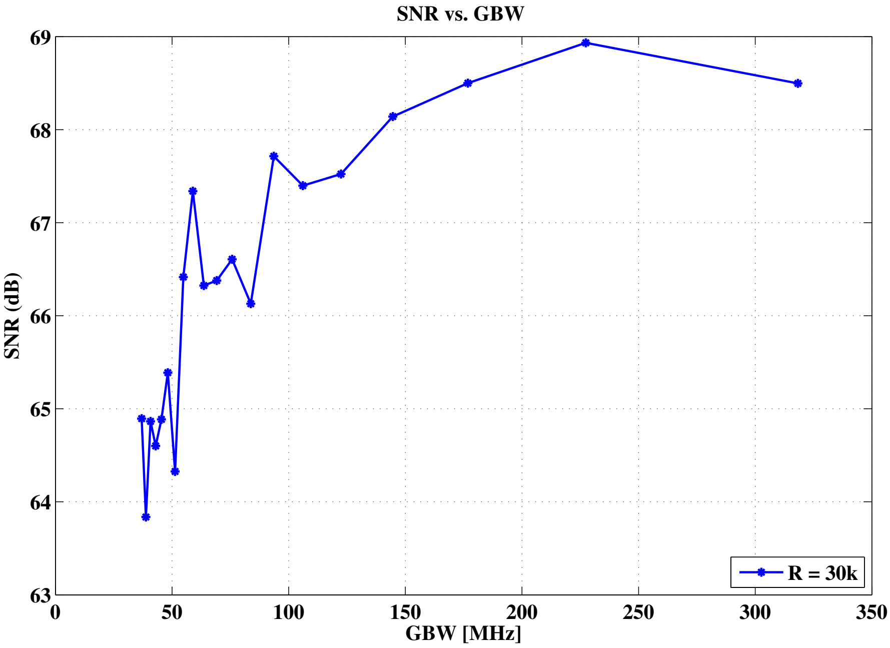


Figure 3.14: SNR varying for different GBW values

The architecture implemented, figure 3.15, is asynchronous to avoid to generate a

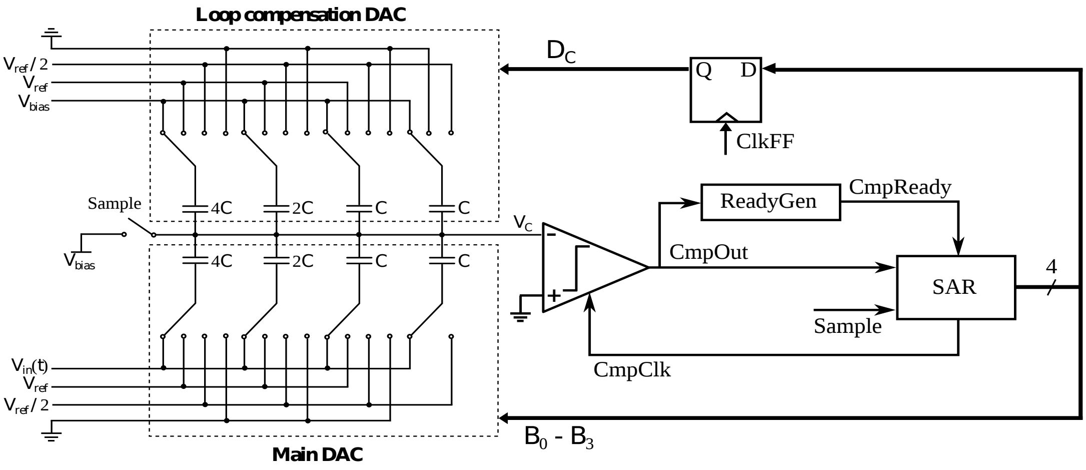


Figure 3.15: Operating scheme of the SAR quantizer

clock 5 times faster than the sampling frequency which will result in more power consumption for the circuit. There are two DACs in the SAR quantizer both implemented using SC technique. The main DAC is used to generate the different voltages during the conversion, the principle is based on charge sharing in a binary weighted capacitors array. The sampling is done on the main DAC before the conversion. The compensation DAC is also implemented here through another binary weighted capacitors array. Having imposed $a^*_{cmp} = 1$ makes the realization of the arrays easier since the two DACs are exactly the same, the right amount of loop delay, $0.56T_s$ in our system, is then given using D flip-flops that delay the output of the modulator, thus the compensation DAC will provide a delayed copy of the output at the comparator input.

Instead of the standard charge sharing scheme a tri-level charge distribution architecture is employed [8]. With this technique the sampling phase is carried out as usual, the top plates of the arrays are connected at V_{bias} while the bottom plates of the main DAC are connected to the input signal, hence sampling the input. During this phase the compensation DAC is reseted by connecting its bottom plates to V_{bias} as well. After the sampling is done the sample switch is opened and the bottom capacitances are connected to $\frac{V_{ref}}{2}$ instead of V_{ref}, this reduces the power consumption from the reference. After the first bit is determined the first capacitor in both DACs is switched then to ground or V_{ref} depending if the bit was 0 or 1. This operation is then carried out for the remaining bits, switching the capacitances between $\frac{V_{ref}}{2}$ and ground or V_{ref} reduces the power consumption and improves the settling time compared to switching between V_{ref} and ground only.

3.6.1 Asynchronous operation

Every conversion step doesn't have a fixed amount of time since the quantizer operation is asynchronous, thus the step to determine the LSB will take longer than the first step to find the MSB since the comparator requires longer times to resolve smaller inputs. The signals for ensuring proper working are implemented in the quantizer. After the sampling phase the SAR logic enables the comparator, when one of the comparator's outputs goes high the *ReadyGen* block detects this and flag it with the *CmpReady* signal. When the *CmpReady* signal is high the comparator is reseted and the DAC update accordingly to the bit value. This goes on until the last bit is determined and then the process stops waiting for the next sampling phase. The *CmpReady* block is implemented with an OR logic gate, figure 3.16, as shown this will generate an high signal when one of the comparator outputs goes high. It is important to ensure the proper operation of this block also when the input at the

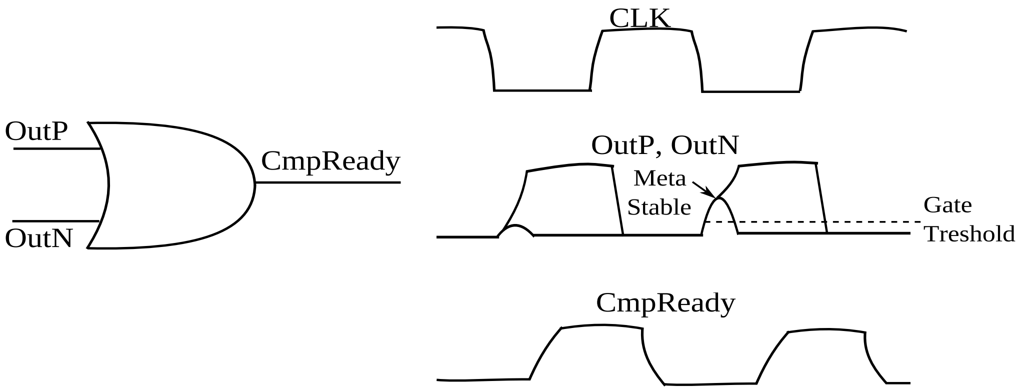


Figure 3.16: ReadyGen logic and waveforms from its working

comparator is low and its outputs are close to the metastable level, waiting for the comparator to resolve its input might slow down too much the process, hence the OR gate has a threshold below the metastable level hence ensuring that the *CmpReady* signal is generated for small signals as well, [15, 39]. A time diagram and the circuit to make it are shown in figure 3.17, this is the logic of the SAR quantizer that allow the asynchronous working of the quantizer. To generate the comparator clock signal,

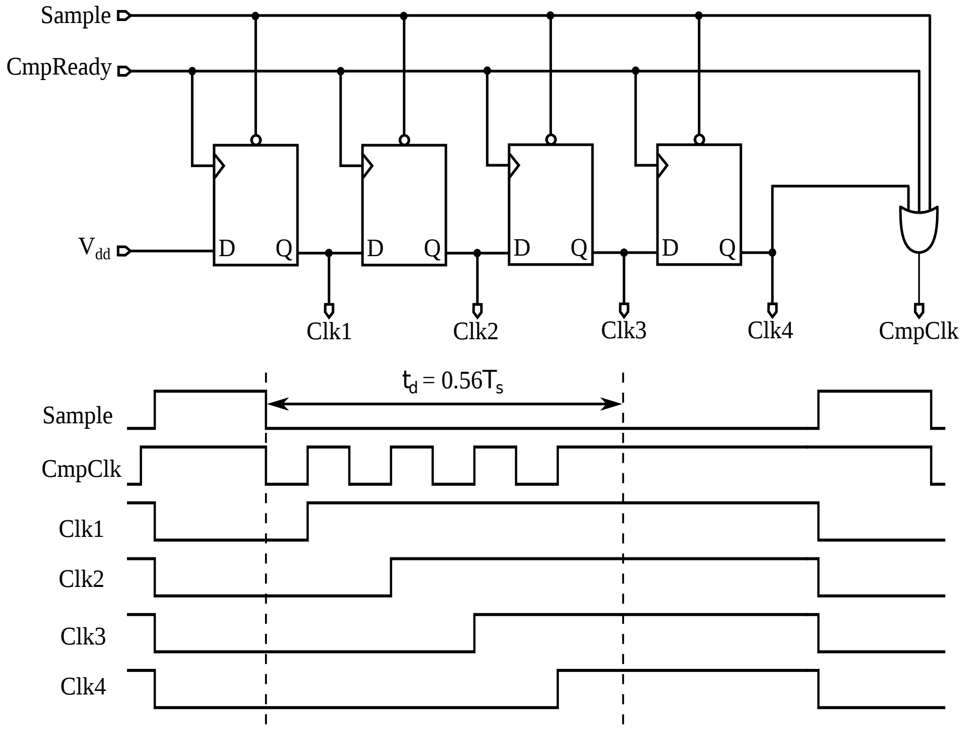


Figure 3.17: Time diagram and logic of the SAR

CmpClk an OR gate is used, the comparator is active low and in reset when the clock is high. The sample signal is generated on another block outside the quantizer (it was actually in the quantizer layout only for it was more convenient to have it there), when the *Sample* signal goes high the comparator clock is high, hence resetting it, thus *CmpReady* is low, also the sample signal resets all the flip-flops generating the *Clki* signals. When the sample is over *Sample* goes low making *CmpClk* to go low too and hence starting the comparison, when the comparison is over *CmpReady* will be high. *CmpReady* acts as the clock of the flip-flops, the only one to update its output is the first one though that has its input connected to V_{DD}, hence $Clk1$ goes high signaling that the first comparison is done while all the others $Clk2$, $Clk3$ $Clk4$ are low. Also when *CmpReady* goes high, *CmpClk* will go high and reset the comparator again, bringing *CmpReady* to a low level also. During this time the switches of the DACs are updated for the next step of the conversion, and the cycle repeats until the last bit. On the last bit $Clk4$ will reset *CmpClk* and signal the end of the conversion, note that without $Clk4$ low, *CmpClk* would go low again after

the reset of the comparator, this signal will instead stop the conversion.
In figure 3.18 the signals from simulation are depicted and the temporal relationship between the control signals and the DACs output is showed. The image refers to

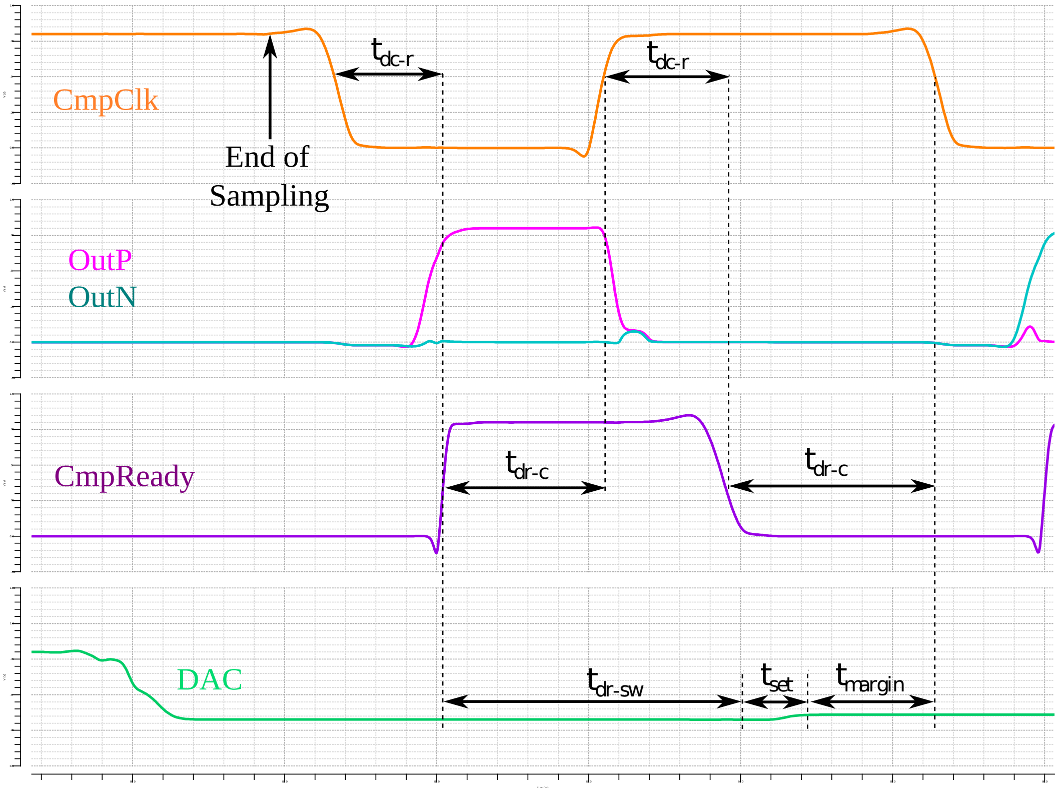


Figure 3.18: Timing scheme of the quantizer

one conversion step, when *CmpClk* is low the comparator is enabled, and one of its outputs will go high triggering *CmpReady* to go high. Then *CmpClk* goes high resetting the comparator and bringing *CmpReady* low which will bring the system into the next conversion step. When *CmpReady* goes high actually two different processes are activated, one will reset the comparator, the other will set the switches of the DACs in the right position for the next comparison. To ensure the proper working of the quantizer when the *CmpClk* goes low again the output of the DAC needs to be updated already or the comparison will yield a wrong value. From the signals can see that the time for the *CmpClk* to go low after the comparison is $2t_{dr-c} + t_{dc-r}$, while $t_{dr-sw} + t_{set}$ is the time required to update the DAC switches and for its output to stabilize, t_{margin} is indeed the margin for correct operation of the quantizer. The condition that must hold true for each comparison is then:

$$2t_{dr-c} + t_{dc-r} > t_{dr-sw} + t_{set}.$$

Extensive simulations were carried out to ensure that this is always verified.
Finally the schematic of the used comparator is showed in figure 3.19 [12]. It consists in two stages, the first one a differential amplifier to suppress the noise from second

stage, the second is a positive feedback latch. This architecture doesn't have any static biasing, hence only dynamic power is consumed, making it suitable for low power applications.

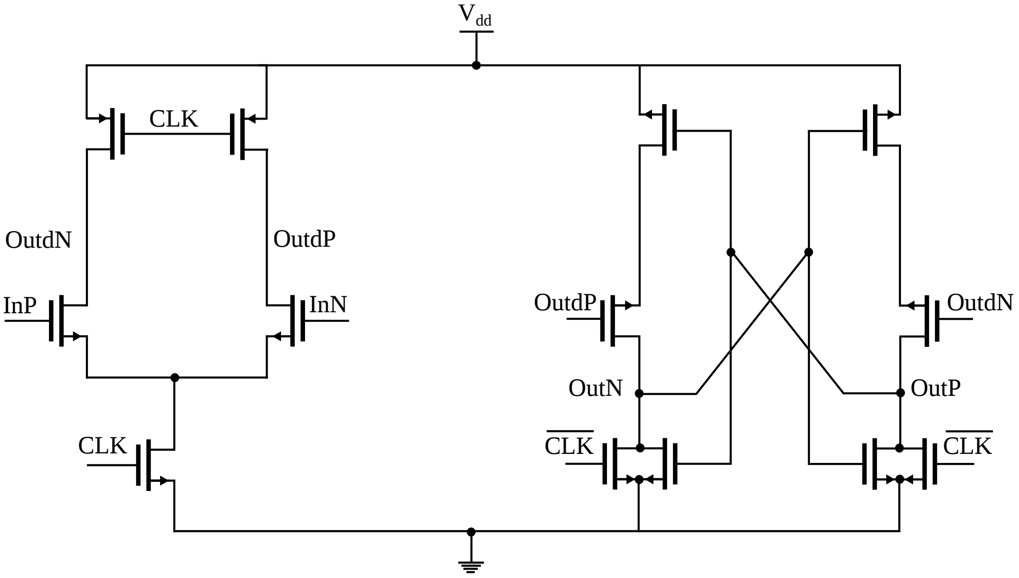


Figure 3.19: Schematic of the comparator used in the quantizer

3.7 ΔΣ Modulator Architecture

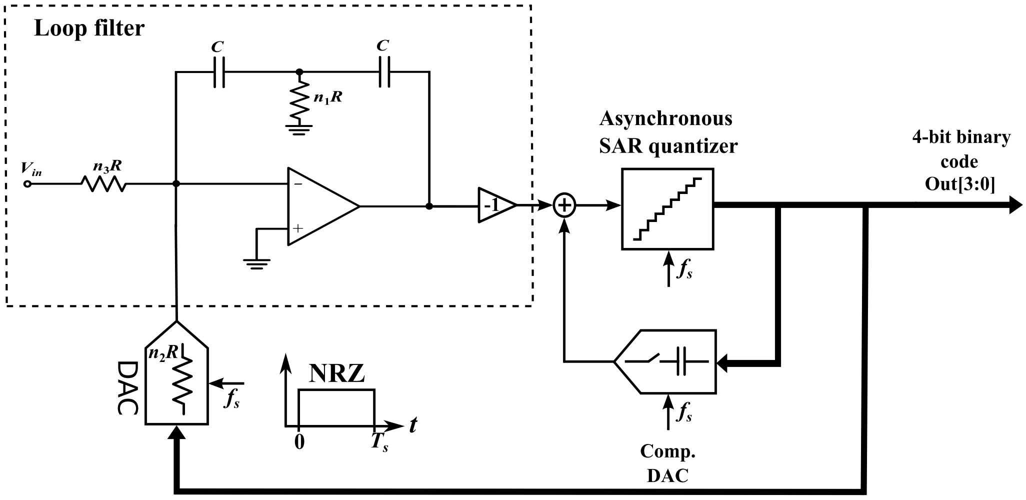


Figure 3.20: Architecture of the implemented modulator

The architecture of the whole modulator is shown in this section and the main choices made for the low power architecture are summarized here.

- $OSR = 16$
  - ▷ Benefits of ΔΣ modulation but f_s is low to reduce power consumption
- Loop Filter: SOA network to reduce number of the opamps
  - ▷ $2nd$ order modulator ⇒ one amplifier needed
  - ▷ Amplifier with feedforward compensation
- Quantizer
  - ▷ 4bit
  - ▷ SAR architecture ⇒ one comparator
  - ▷ Tri-level SC DACs

Chapter 4

Simulations results and former circuit

In this chapter the simulations to verify the performance of the circuit are presented with the results achieved, the organization is as follow:

- The first section will cover the results at schematic level
- In the second part the previous circuit is presented with the problems found at post layout simulation level, the adopted strategy to solve the issues is finally explained along with showing post layout simulations of the whole circuit.

4.1 Simulation results of the circuit

4.1.1 Amplifier and network simulations: frequency response and stability

In this section the most relevant results from the amplifier simulations are shown. Figure 4.1 the open loop frequency response of the amplifier for differential mode is shown. The DC gain is $66dB$ which as shown in section 3.5.1 is enough to achieve the desired performance and also to have the poles of the SOA network close enough to the origin. Unity frequency $f_u = 216MHz$ is also more than the minimum requirements found in 3.5.1. The biasing current of the amplifier is $52\mu A$. We can also note that in open loop configuration this amplifier has a $PM = 62°$, when looking at the common mode response this was found to be unstable, figure 4.2, and its compensation was made when the amplifier is placed in the network, as compensating also for the open loop configuration was unnecessary, since the amplifier is only to be used with the SOA network and doing so resulted in not achieving the minimum requirements for the modulator to work. The poles and zeros of the amplifier in open loop configuration are shown in figure 4.3, as can be seen there are two poles with positive real part, hence the system is unstable in open loop configuration. In figure 4.4 is a zoomed section of the poles and zeros showing the zero introduced by the feedforward path compensating one of the amplifier poles.

The DM loop gain of the SOA network and the implemented amplifier is shown in figure 4.5.

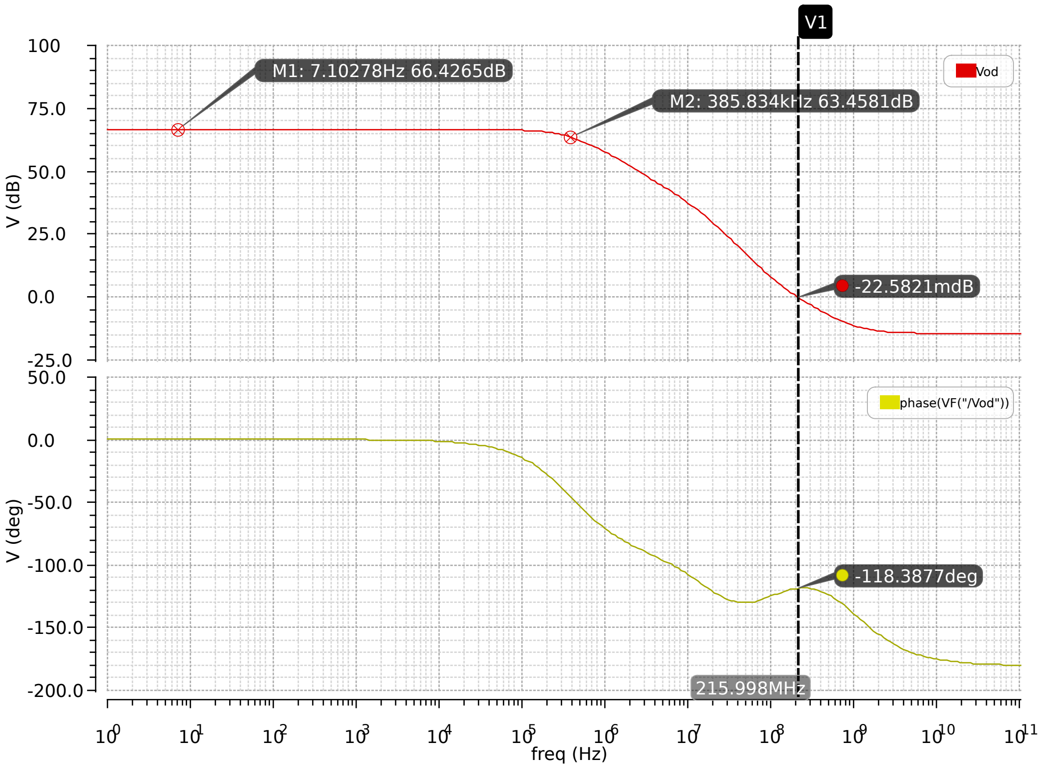


Figure 4.1: Open loop frequency response of the amplifier

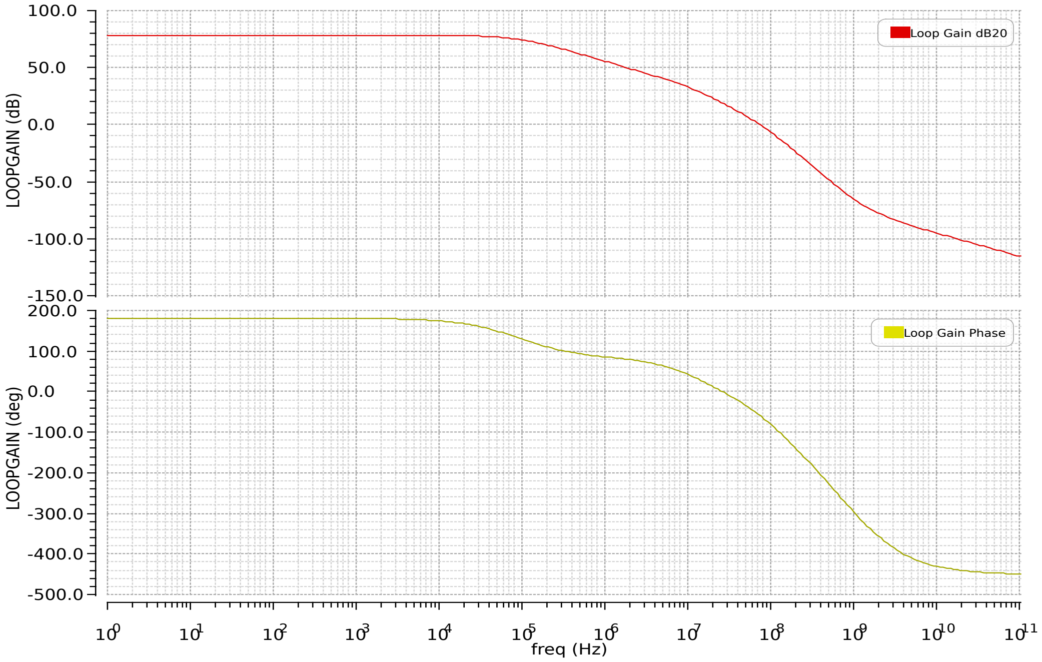


Figure 4.2: Open loop frequency response of the amplifier for CM signals

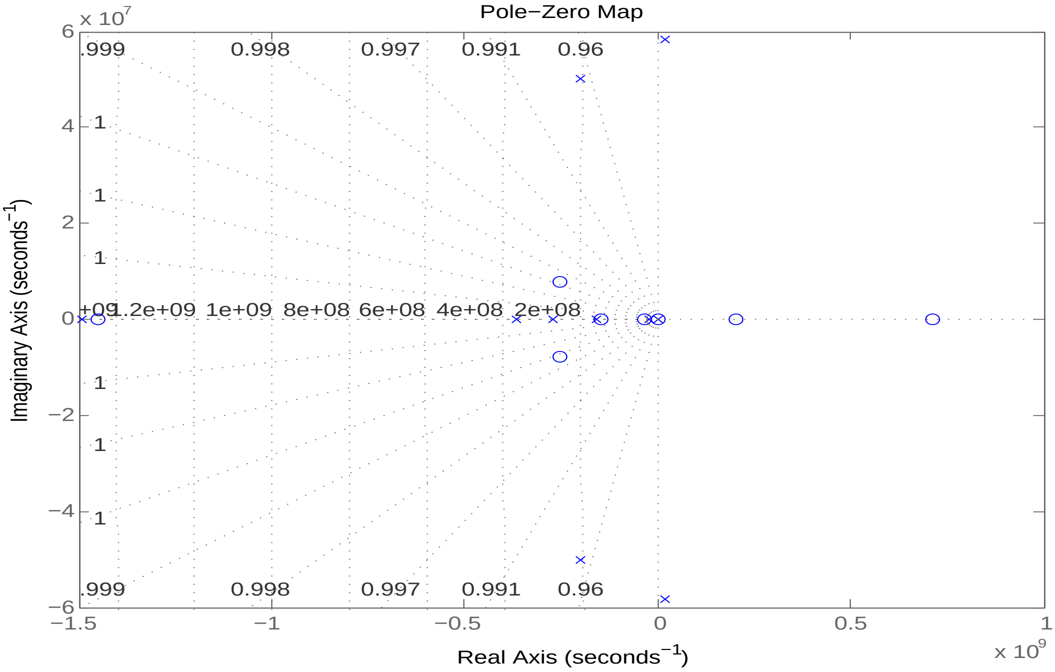


Figure 4.3: Open loop poles and zeros of the amplifier

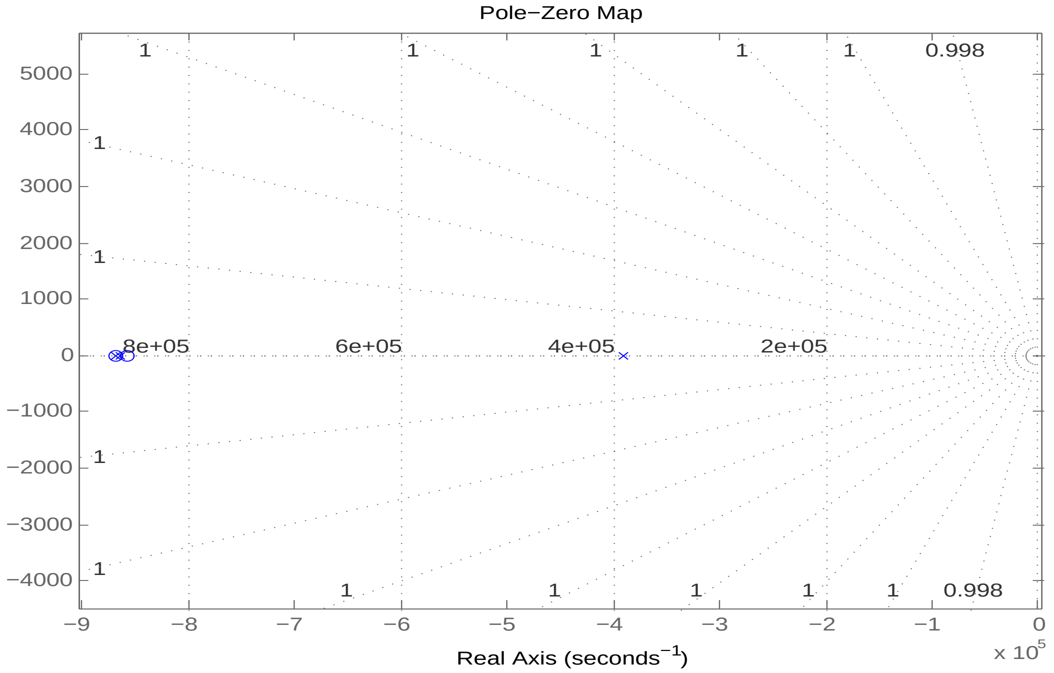


Figure 4.4: Feedforward compensation in the realized amplifier

The PM can be calculated using the markers on the graph and is found to be $PM = 80°$. To ensure proper working of the system different corners simulation were ran, the results for stability are summarized in the table below.

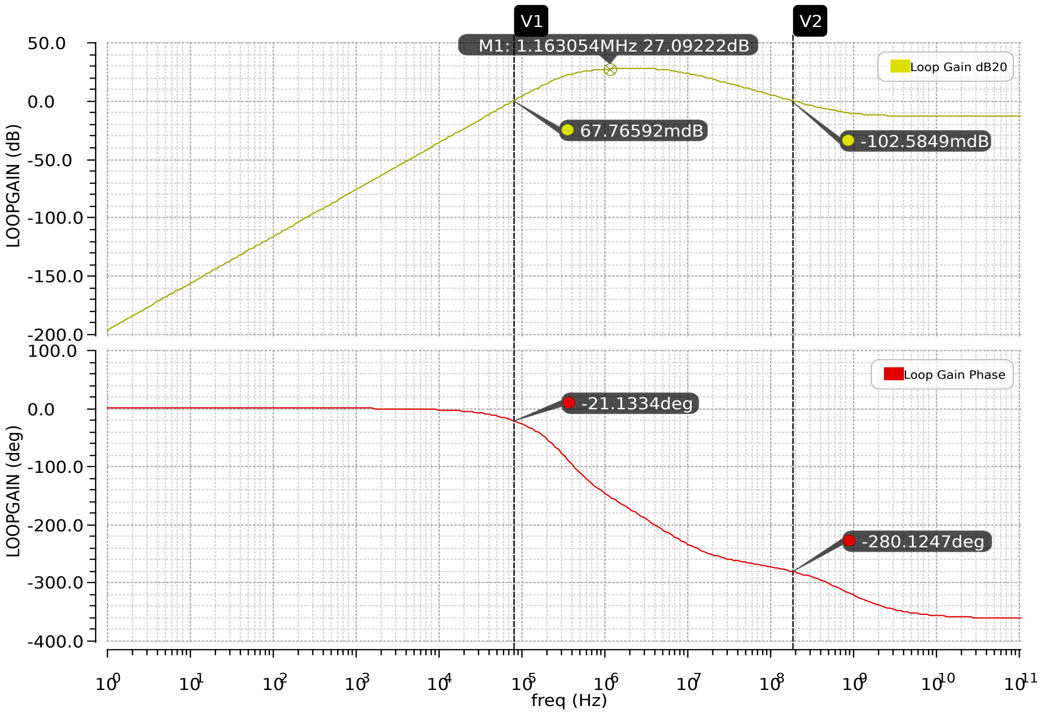


Figure 4.5: Loop gain and phase for DM of the loop filter

Corner	PM
TT	80.07°
FF	72.95°
FS	94.09°
SF	72.99°
SS	90.81°

Where in the corner field the first letter refers to the n-mosfets and the second one refers to p-mosfets, e.g. FS means fast n-mosfets and slow p-mosfets, also TT refers to typical typical which sets indeed the typical parameters of the technology.

The analysis will be now focused on the CM operation of the loop filter, before showing the simulation results it is worth to mention that when we close the amplifier in feedback we have two different loops for the CM, one is the CMFB loop internal to the amplifier and another one is the one through the feedback network, it was shown in [25] that in such a configuration to determine the stability of the CM loop it is possible to measure only one loop gain using return ratio in a point that breaks both loops. Hence reporting one measure will tell about the stability of the whole system.

In figure 4.7 the loop filter response for the CM is shown. The DC gain is quite high, having a DC gain too high is usually a problem for the common mode loop as it will cross the x axis later when the non dominant pole is giving almost full contribution to the phase and making the system unstable. This is not the case for this system were a zero on the CM loop compensate the system and having an high DC gain helps to keep a CM value at the output close to the reference in all the

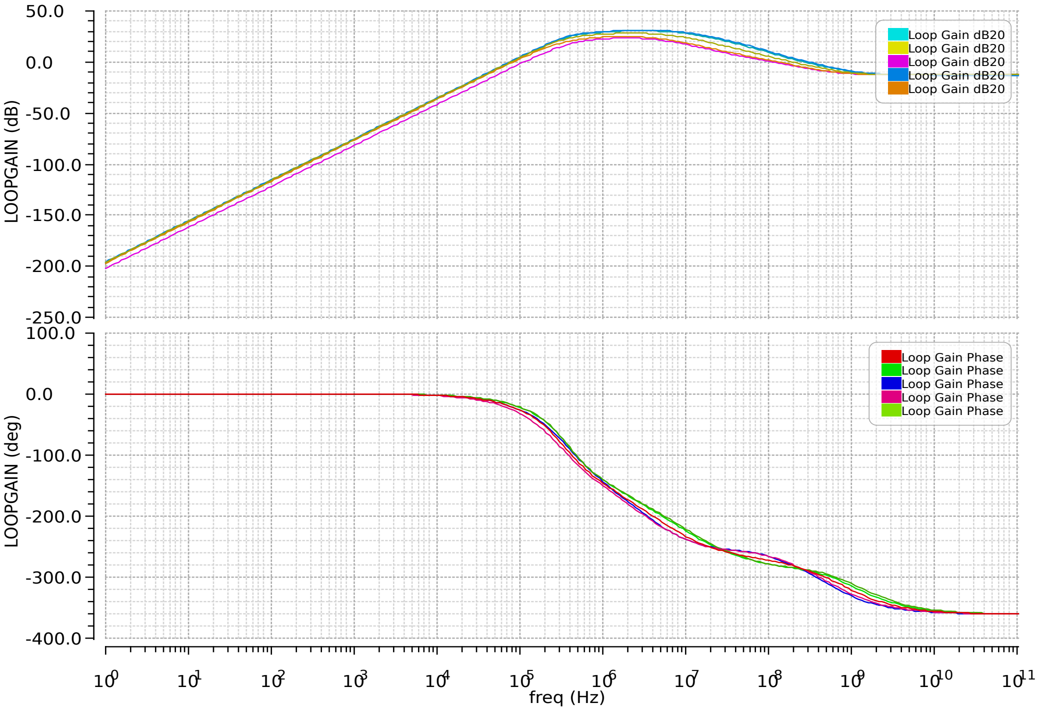


Figure 4.6: Loop gain and phase for different corners simulations

corners scenarios. It is also worth noticing that the loop gain crosses the $0dB$ axis more than one time, to be sure of the stability the Nyquist's plot was traced, and it's showed in figure 4.8, and a zoom close to the -1 point is shown in figure 4.9. As can be seen the plot doesn't encircle the -1 point and is stable. Also in figure 4.10 the poles and zeros of the loop filter are shown in the worst case for stability to ensure that the two poles with positive real part are now moved into the left part of the pole zero plot. Below, as done for the DM case, a table summarizing the stability of the system for the different corners is reported. For the FS scenario the PM has a big drop compared with the other cases.

Corner	PM
TT	96.97°
FF	96.78°
FS	65.87°
SF	92.76°
SS	78.8°

The step response to a variation of the CM reference is also shown in figure 4.11, where V_{cm} is the reference. It can be seen that the CM output follow the reference very closely ($399mV$ and $408.9mV$ with $400mV$ and $410mV$ reference). The peaking shown in the step response is unusual when compared with the PM of the loop this is due to the network poles, in fact using an ideal amplifier with high gain the peaking become an oscillation, since the amplifier used was an ideal gain block this effect is due to the poles of the network.

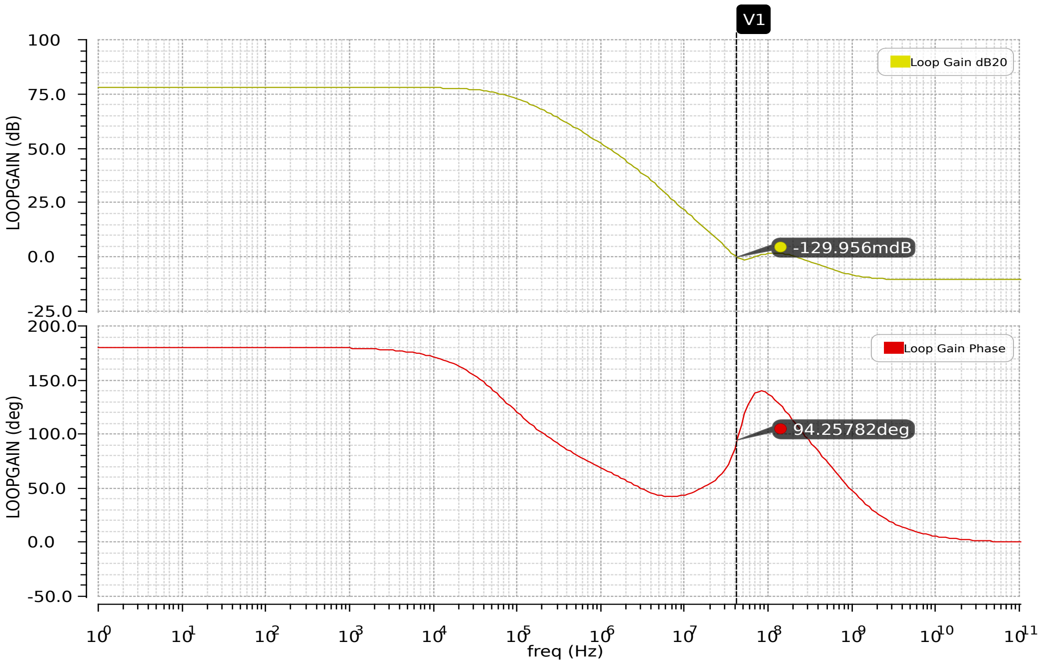


Figure 4.7: Loop gain and phase for CM signals of the loop filter

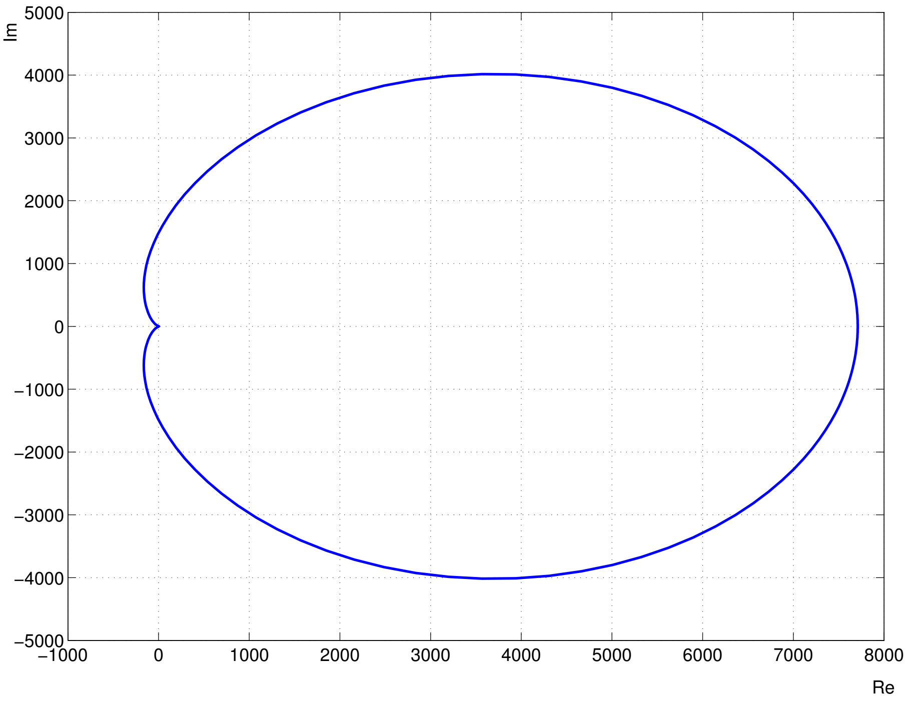


Figure 4.8: Nyquist's plot for the CM loop

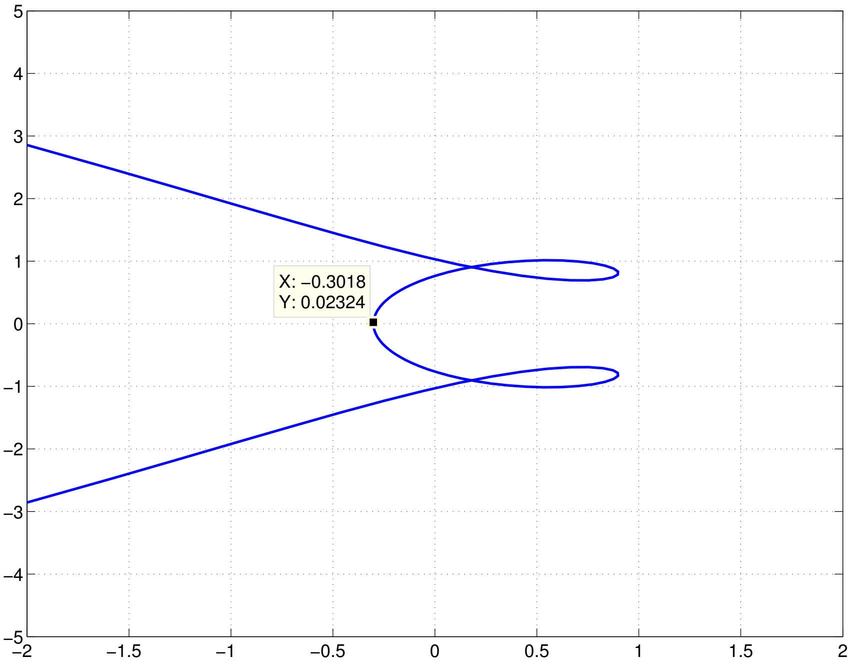


Figure 4.9: Zoom of the Nyquist's plot on the -1 point

4.1.2 Modulator performances

After looking at the frequency response and stability of the amplifier and the loop filter simulations to test the performance of the whole circuit were also made. Figure 4.12 show the modulator output along with the loop filter output and the input of the system. In figure 4.13 the output spectrum of the modulator is shown and compared with the matlab model. For a differential input of $200mV$ the modulator achieve a $SNR = 65.3dB$ consuming only $69\mu W$ per conversion. Several simulations varying the input amplitude were made to plot the dynamic range of the modulator, figure 4.14.

4.2 Previous circuit version

This project was started by a former PhD student at Lund University, Dejan Radjen who made this very modulator but with another SOA network and different amplifier, the circuit that was fabricated had stability issues. In previous circuits the loop filter and DAC worked fine while this was a first attempt at using a SAR quantizer in the modulator. It was found that indeed the quantizer was making the modulator unstable. The measured circuit was unstable, its output swinging from ground to V_{dd}, since this problem arose only in the circuit but wasn't seen in schematic level simulations the best hypothesis was that the compensation DAC was heavily affected by parasitic capacitances, which are the most relevant effect when comparing layout with schematic for this kind of modulator, thus not having the right compensation

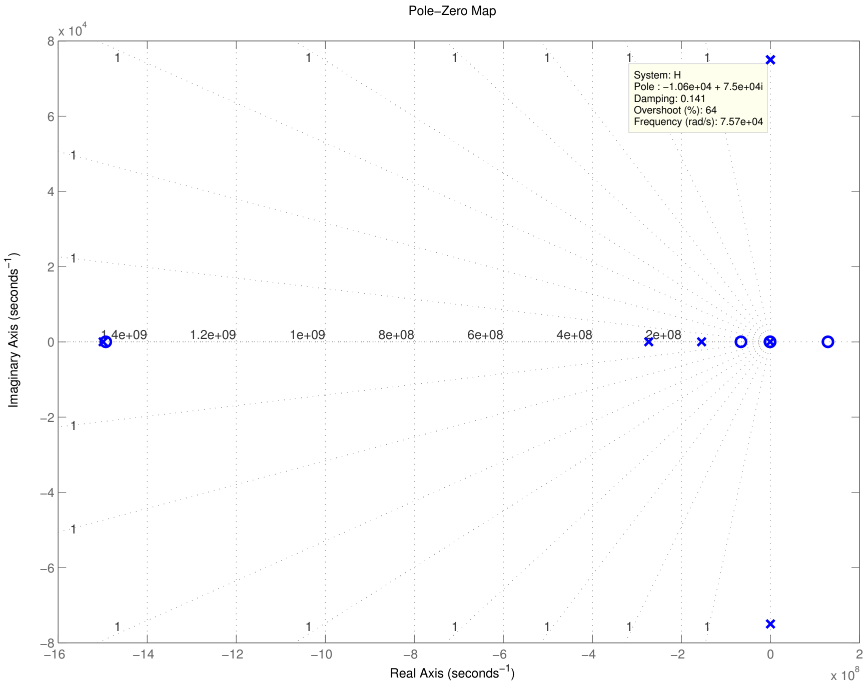


Figure 4.10: Poles and zeros of the SOA network for the FS corner case

the circuit would become unstable. In figure 4.15 the output signal of the modulator for a zero input is shown, the loop filter and DAC models were schematic level models while for the SAR quantizer a post layout model was used. The post layout model takes into account the parasitic contribution of the metal connections, namely the resistances of the path and the capacitances both from the path and from the close by metal connections. As can be seen the simulation confirms the result from the fabricated chips that including the parasitics in the circuit made it unstable. At the time there was no time to ran a post layout simulation since it's very time consuming, also being able to review the issue at simulation level make it possible to solve it, or we couldn't have a feedback if the issue is fixed or still there. As can be seen in figure 4.15 the output swings from ground to the voltage reference of the quantizer, an ideal DAC was used to convert the bits into an analog signal for showing the output and its reference voltage was set to be equal to the quantizer, and also all the 4bits of the quantizer are shown and they are randomly changed, even when the input is zero.

The first thing that was checked was the gain on the compensation path as this was the best clue to why the modulator wasn't working, the values found where:

Schematic	$a^*_{cmp} = 0.8$
Post Layout	$a^*_{cmp} = 0.787$

Hence the compensation path wasn't the problem for the instability, even if the value is different from the calculated one the performance of the system are not affected by it.

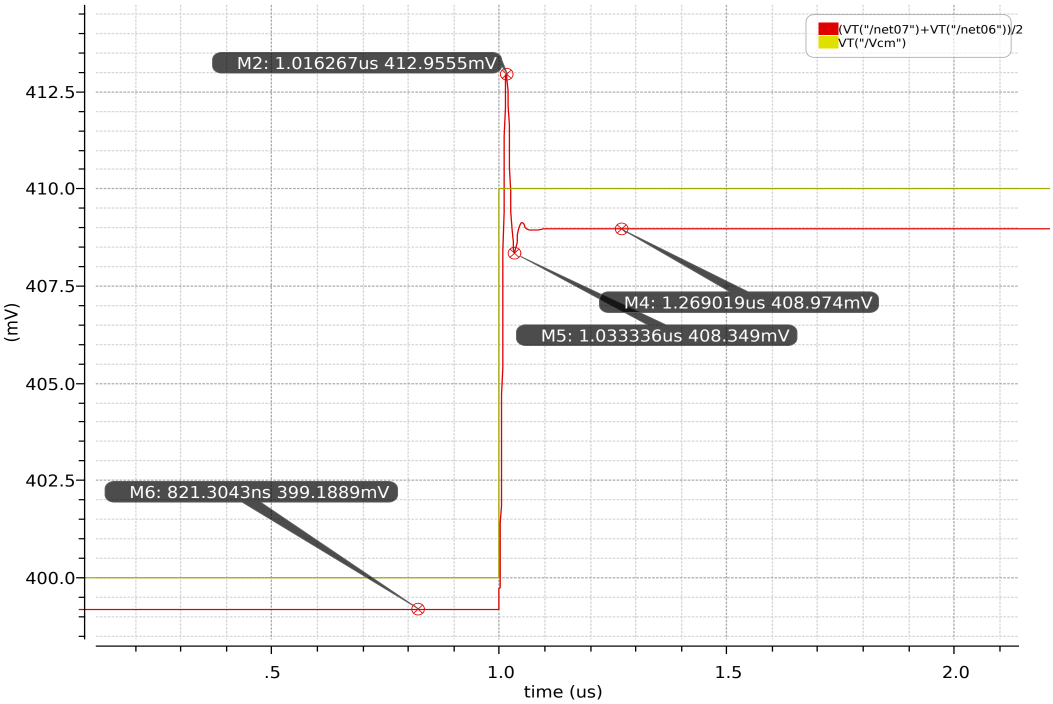


Figure 4.11: Step response of the CM output for a variation of the reference

To find the issue with the circuit all the sub-blocks making the SAR quantizer were one by one substituted by the post layout model to find if one of them was causing the problem. Unfortunately changing all the blocks inside the quantizer to use the post layout model gave a working quantizer, hence the problem had to be when connecting two or more blocks together. The two critical blocks are the SAR logic(figure 3.17) and the DACs in the quantizer. When connecting them together the post layout simulation revealed the instable system. It was nailed down that when connecting these two blocks together the value at the comparator changed from schematic to post layout. Figure 4.16 shows how the input of the comparator changed between the two simulations, this discrepancy made the feedback system unable to recover leading to instability.

4.3 Resolving the layout related issue

The true reason for the DACs to have different values when simulating them with post layout models was found to be a glitch with the flip-flop in the SAR logic. Especially for low input the comparator latch was too slow and caused one of the flip-flops to generate an high value at the output where it should have been low and the other way round also. This is critical for the working since the circuit was implemented with differential architecture and the glitch appeared only at the negative signal. Hence the values at the comparator inputs were different from the schematic and also not differential signals, since one bit was correct and only the negative was wrong. In fact the output of the comparator was found to be more slow

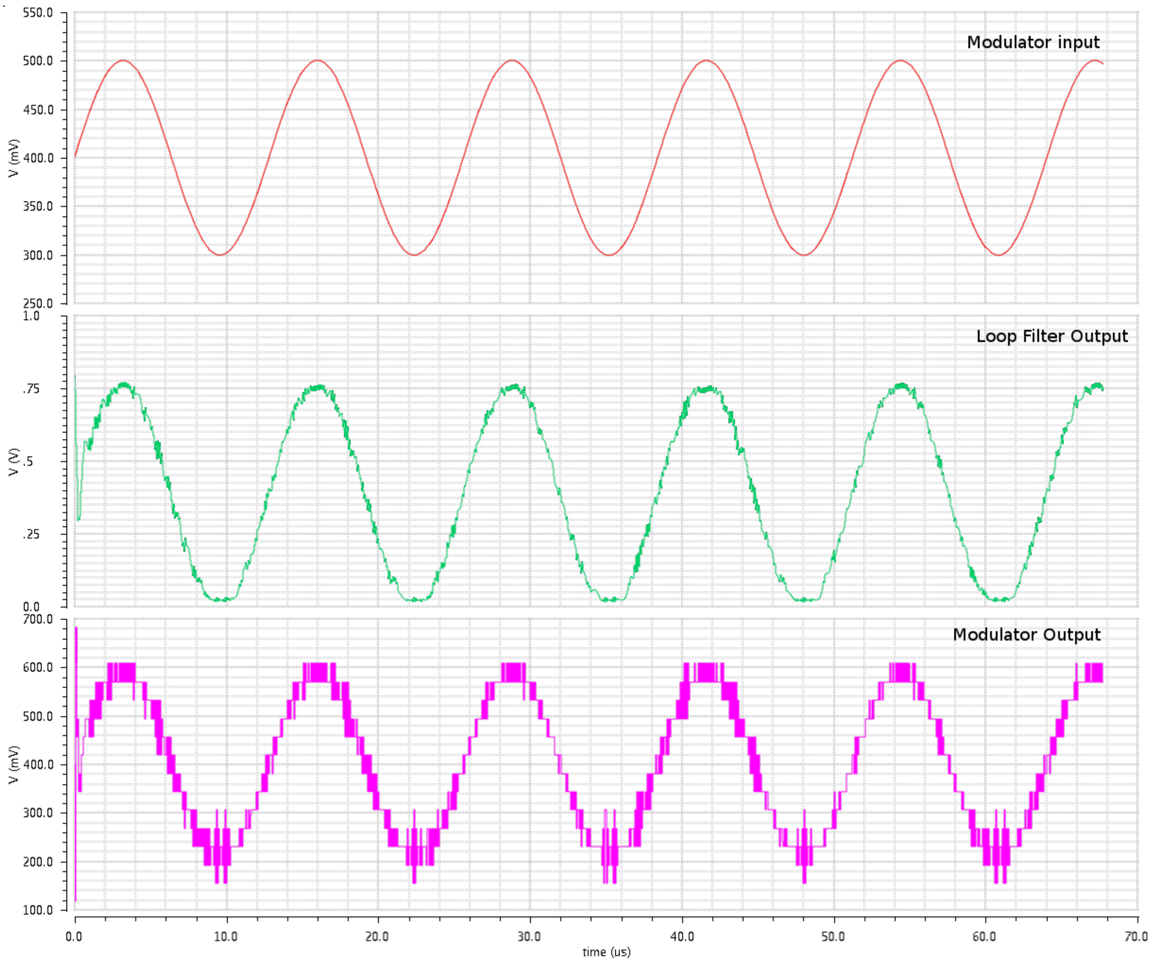


Figure 4.12: Input signal and outputs from the modulator and the loop filter

for the negative output and the glitch was only on the negative part of the differential signal. The slowness of the latch was due to the transistors sizing used, $\frac{W}{L} = 8$ and $\frac{W}{L} = 5$ for pmos and nmos respectively, while this was working on schematic level this sizing wasn't enough to drive the parasitic capacitances effectively to keep proper functioning of the system. The comparator was then redesigned increasing its mosfets and making it able to properly drive its load with the parasitic capacitances. Since small transistors were used inside the SAR logic the critical paths were found and their temporal margin increased to ensure proper operation.

For instance in figure 4.17 the margin for the second step of the conversion is shown, in figure 4.18 this is improved. The margin is from when the DAC output is at the correct value for that conversion step to when the comparator actually start the comparison, as can be seen with the first version this was about $300ps$, improving the speed of the chain that brings the command to the switch made it possible to achieve a margin of about $800ps$.

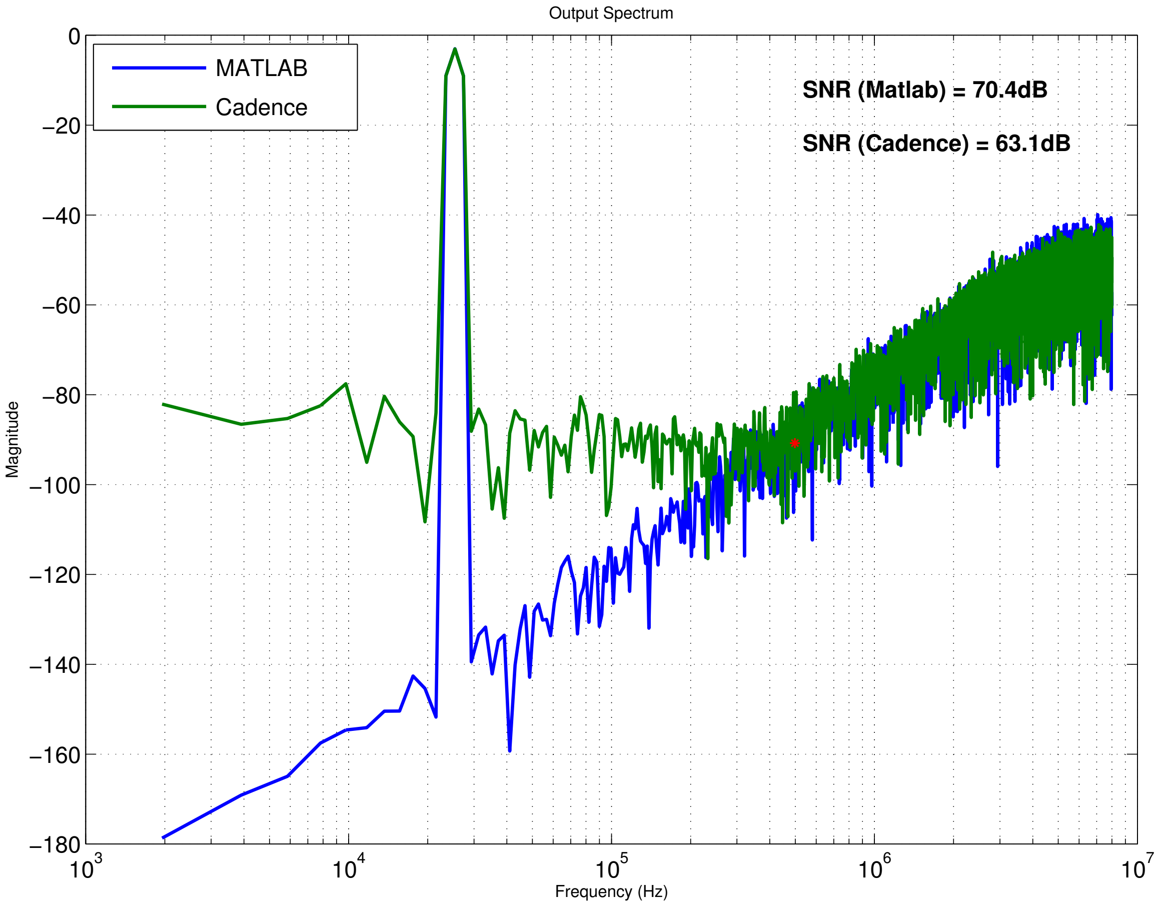


Figure 4.13: Output spectrum and SNR of the modulator with a $200mV$ differential input, compared with the ideal matlab model

4.4 Post Layout simulation results

After increasing the latch speed the quantizer was properly working already in post layout simulations, to ensure this the critical paths were analyzed and in case their timing improved by increasing the buffers that are at the end and at the beginning of the paths, this was usually enough to achieve good results. When this wasn't sufficient a redesign of that path was necessary. After achieving a good result overall we also ran a simulation at 105°, increasing the temperature in fact slows down all the mosfets hence being critical for the timing of the quantizer. A simulation of the modulator at 105° was made, using the post layout model for the quantizer, this kind of simulations are high demanding in terms of computational power and hence there was little time to ran longer ones. The output is shown in figure 4.19, as can be seen the output has more noise than the normal schematic simulation but it's clearly not unstable, which was our goal for this simulation. From the output the spectrum can be calculated we can see that the performances of the modulator are lower than the ones at room temperature and the SNR drops to $52dB$, still this was

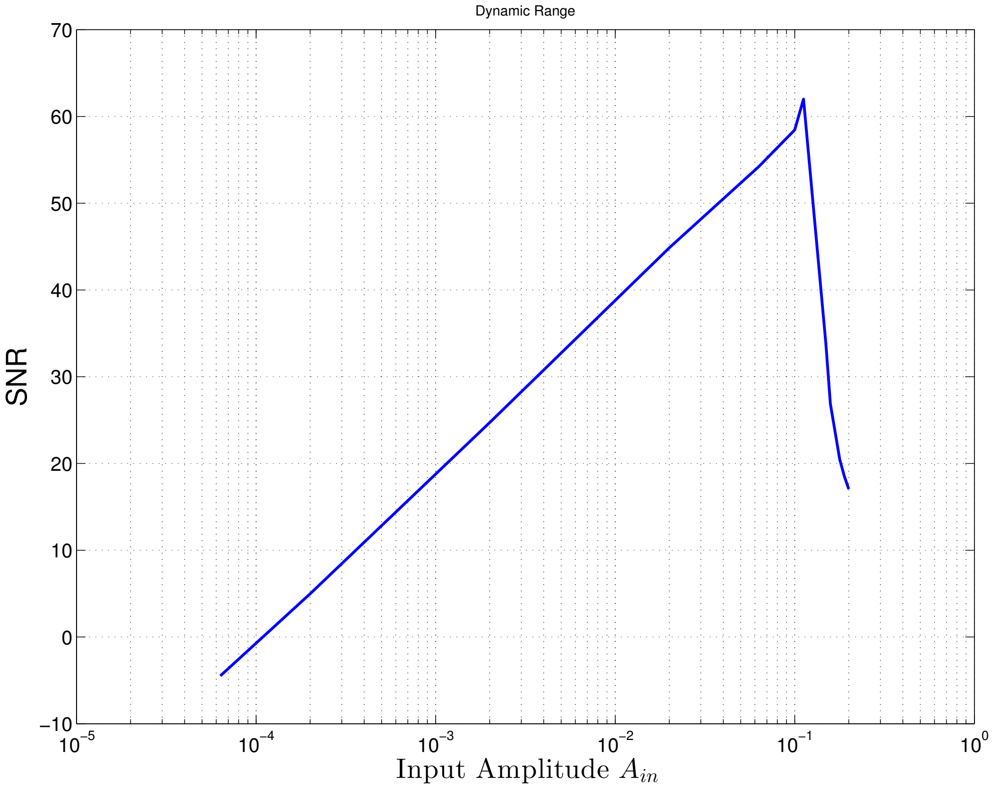


Figure 4.14: Dynamic range of the modulator

a test to ensure the modulator will work at such temperature not that it will achieve its full performance. Still from figure 4.20 we can see that the modulator is properly shaping the noise.

4.5 Performance summary

We briefly summarize the specs of the circuit and its performances. A comparison between other circuits could be made but since the majority of results found in literacy refers to measured circuit a comparison would not be fair since the circuit was only simulated in this work.

V_{dd}	$800mV$
Bandwidth	$500kHz$
Fs	$16MHz$
OSR	16
SNR	$65.3dB$
Power	$69\mu W$
Active Area	$0.09mm^2$

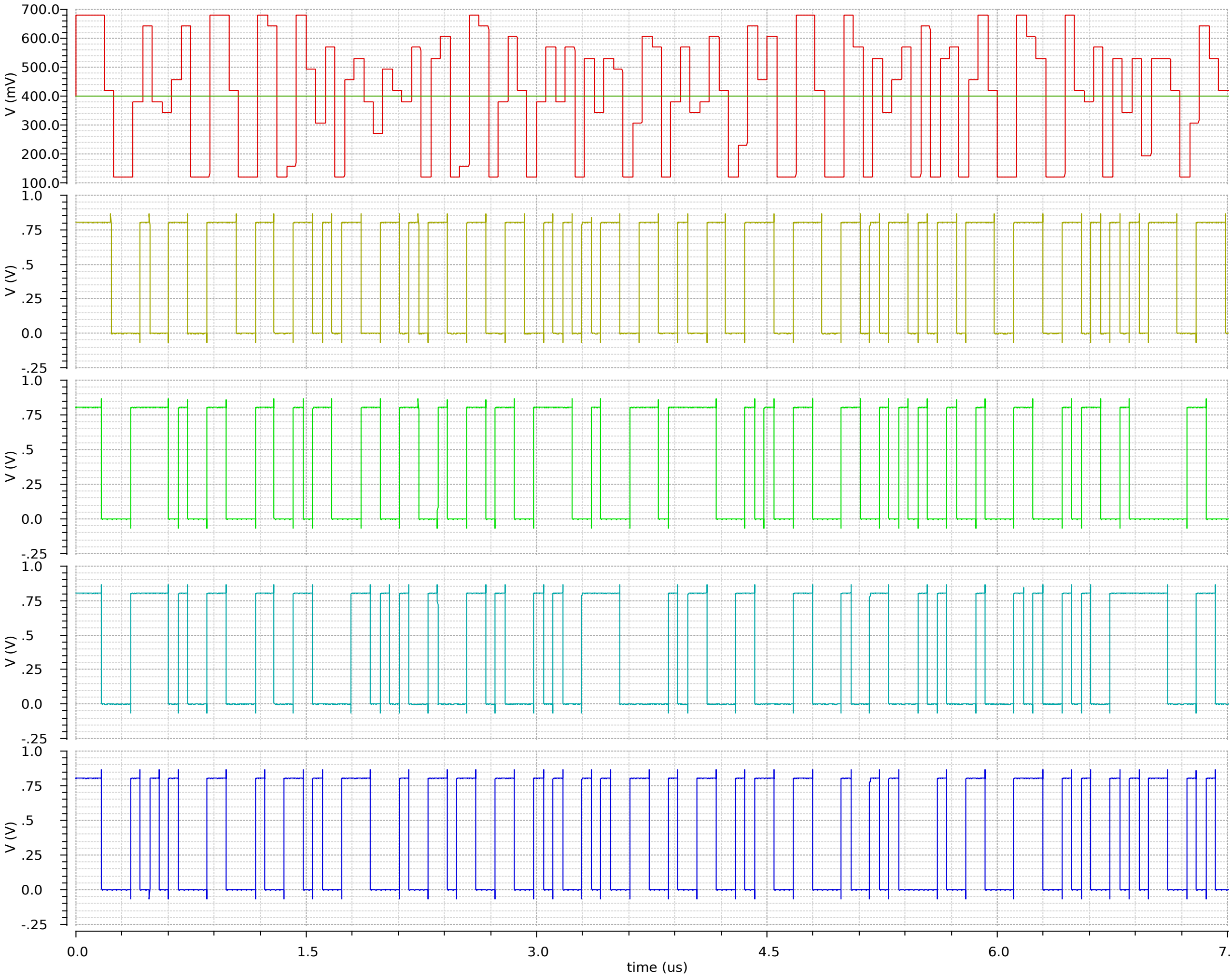


Figure 4.15: Modulator's output with SAR quantizer in post layout simulation showing the modulator output and zero input(only common mode voltage) and then the 4 output bits

4.6 Future developments

The layout of the modulator was finished and the chip will be fabricated again and measured. Below is the layout of the whole chip, including pins and decoupling capacitances that will be sent to the foundry. On the top part of the layout a buffer circuit was implemented to bring the outputs of the modulator out to the pads which are all around the 3 sides of the chip. The modulator is in the center of the bottom part and all around it are decoupling capacitances.

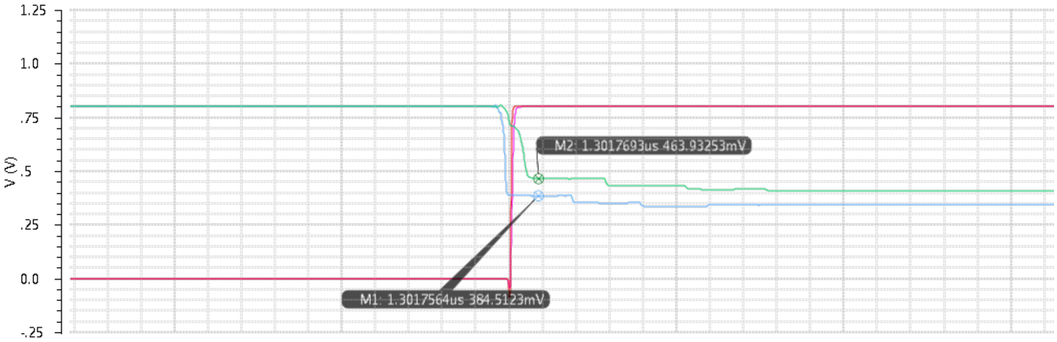


Figure 4.16: Different values at the comparator input for schematic and post layout simulation

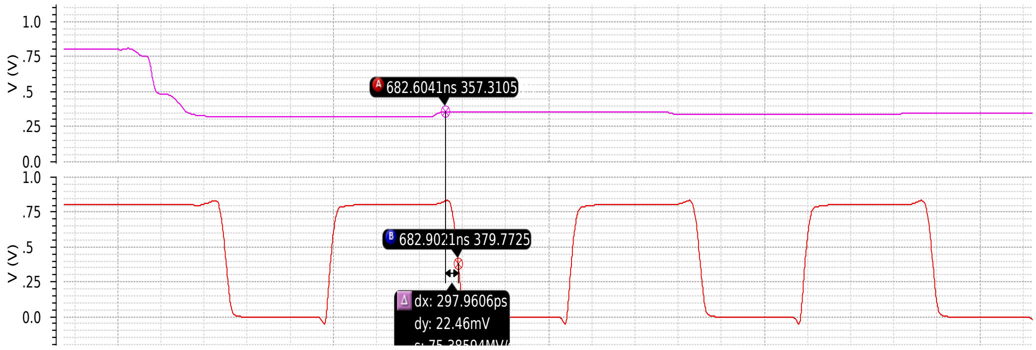


Figure 4.17: Margin for the proper working of the quantizer

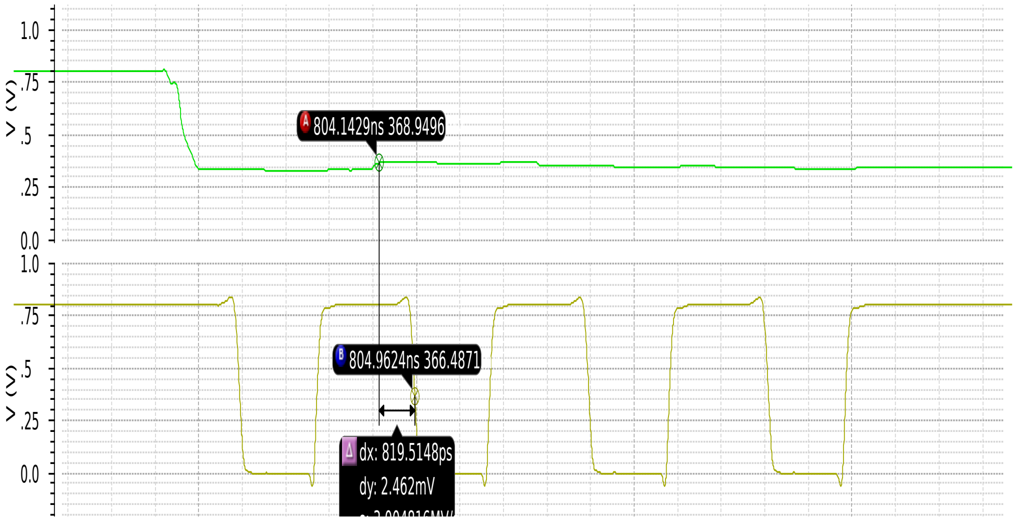


Figure 4.18: Margin for the proper working of the quantizer

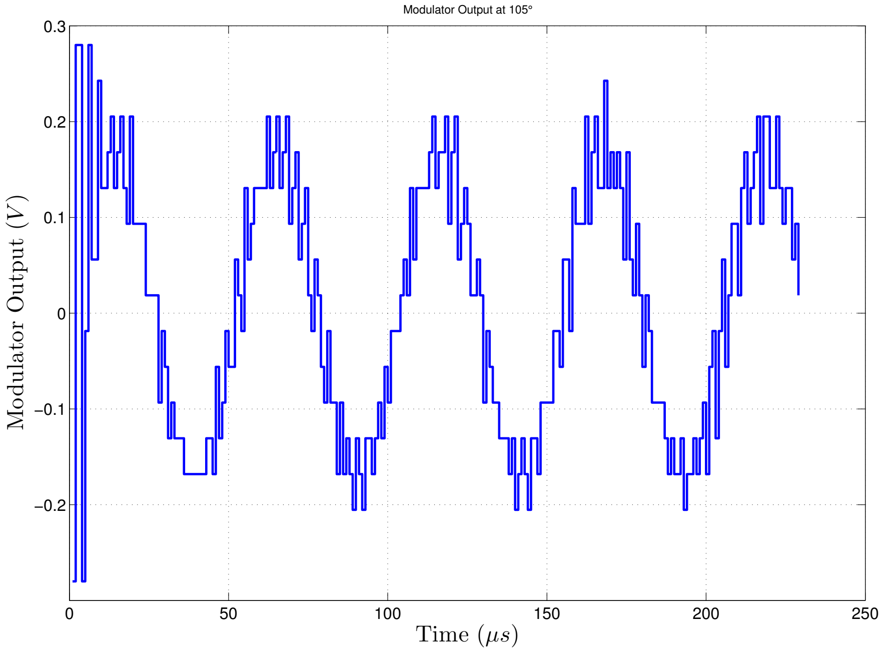


Figure 4.19: Output of the modulator at 105°

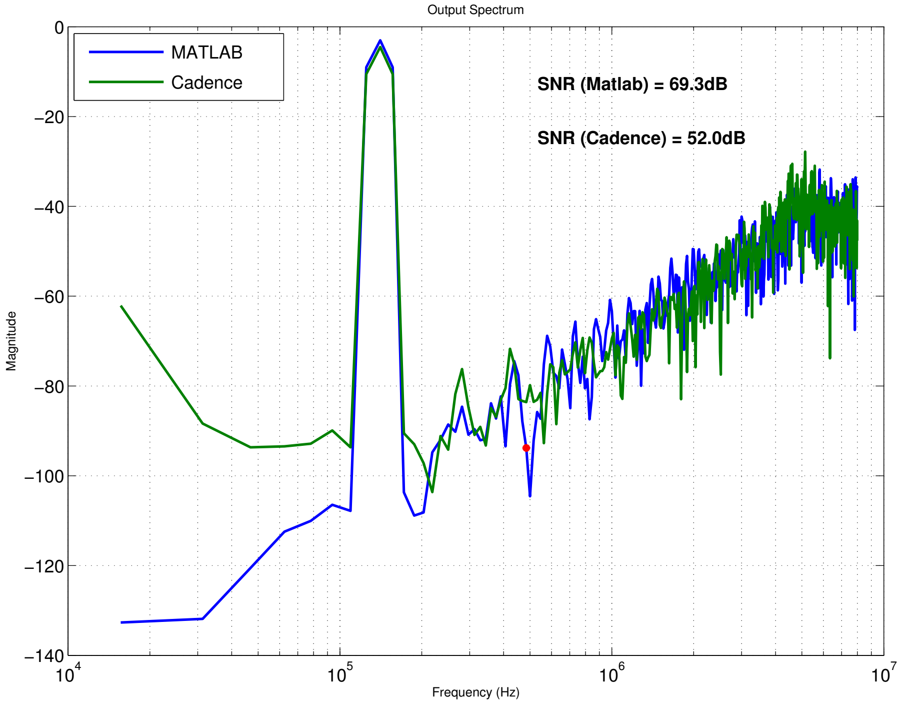
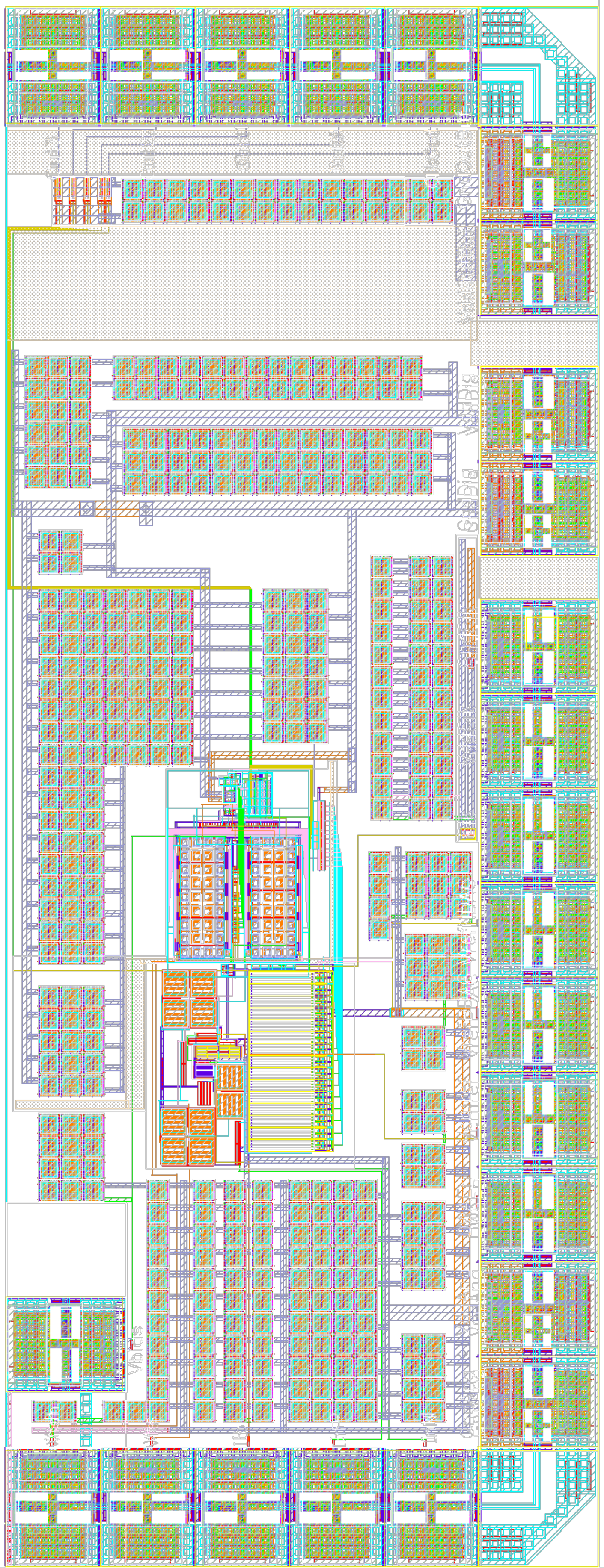


Figure 4.20: Spectrum of the output of the modulator at 105° and the ideal matlab system


Vbias

Bibliography

[1] D. L. Snyder A. B. Sripad. “A Necessary and sufficient condition for quantization errors to be uniform and white”. In: *IEEE Trans. Acoust., Speech, Signal Processing* (Oct. 1997).

[2] M. Anderson. “CMOS Analog-to-Digital Converters - Analysis, Modeling and Design”. PhD thesis. Lund University, 2008.

[3] Mattias Andersson et al. “Theory and design of a CT$\Delta\Sigma$ modulator with low sensitivity to loop-delay variations”. In: *Analog Integrated Circuits and Signal Processing* 76.3 (2013), pp. 353–366.

[4] “Approaches to simulating continuous-time delta sigma modulators”. In: *IEEE Int. Symp. Circuits Syst.* (June 1998).

[5] I. Kollár B. Widrow and M.-C. Liu. “Statistical theory of quantization”. In: *IEEE Trans.Instrm. Meas* (Apr. 1996).

[6] Philippe Benabes, Mansour Keramat, and Richard Kielbasa. “A methodology for designing continuous-time sigma-delta modulators”. In: *Proceedings of the 1997 European conference on Design and Test*. IEEE Computer Society. 1997, p. 46.

[7] Yuan-Shuo Chang et al. “An Analytical Approach for Quantifying Clock Jitter Effects in Continuous-Time Sigma Delta Modulators”. In: *Circuits and Systems I: Regular Papers, IEEE Transactions on* 53.9 (2006), pp. 1861–1868.

[8] Yanfei Chen, Sanroku Tsukamoto, and Tadahiro Kuroda. “A 9b 100MS/s 1.46 mW SAR ADC in 65nm CMOS”. In: *Solid-State Circuits Conference, 2009. A-SSCC 2009. IEEE Asian*. IEEE. 2009, pp. 145–148.

[9] James A Cherry and W Martin Snelgrove. “Clock jitter and quantizer metastability in continuous-time delta-sigma modulators”. In: *Circuits and Systems II: Analog and Digital Signal Processing, IEEE Transactions on* 46.6 (1999), pp. 661–676.

[10] *Delta-Sigma Data Converters*. IEEE Press, 1997.

[11] *Design of Multi-Bit Delta-Sigma A/D Converters*. Kluwer Academic Publisher, 2002.

[12] Michiel van Elzakker et al. “A 10-bit charge-redistribution ADC consuming 1.9 W at 1 MS/s”. In: *Solid-State Circuits, IEEE Journal of* 45.5 (2010), pp. 1007–1015.

[13] “Excess loop delay in continuous-time delta-sigma modulators”. In: *IEEE Transactions on Circuits and Systems* (Apr. 1999).

[14] Y. Yasuda H. Inose and J. Marakami. “A telemetring System by Code Modulation, Delta-Sigma Modulation”. In: (Sept. 1962).

[15] Pieter JA Harpe, Cui Zhou, and Yu Bi. “Nick P. van derMeijs, Xiaoyan Wang, Kathleen Philips, Guido Dolmans, Harmke de Groot,“A 26 μW 8 bit 10 MS/s Asynchronous SAR ADC for Low Energy Radios””. In: *IEEE J. Solid-State Circuits* 46.7 (2011).

[16] Luis Hernández et al. “Modelling and optimization of low pass continuous-time sigma-delta modulators for clock jitter noise reduction”. In: *IEEE International Symposium on Circuits and Systems*. 2004, pp. 1072–1075.

[17] Nedved J. Vanneuville J. Gevaert D. Sevenhans J. “A transistor-only switched current sigma-delta A/D converter for a CMOS speech codec”. In: *IEEE J. Solid-State Circuits* (July 1995).

[18] P.Rombouts J.D. Maeyer and L.Weyten. “Efficient Multibit Quantization in Continuous-Time $\Delta\Sigma$ Modulators”. In: *IEEE Transactions on Circuits and Systems* (Apr. 2007).

[19] Liyuan Liu et al. “A 92.4 dB SNDR 24kHz $\Delta\Sigma$ modulator consuming $352\mu W$”. In: *Low Power Electronics and Design (ISLPED) 2011 International Symposium on*. IEEE. 2011, pp. 351–356.

[20] et al. (2013) Luo H. “A 0.8-V-230-lW98-dB DR inverter-based delta-sigma modulator for audio applications”. In: *IEEE J. Solid-State Circuits* (2013).

[21] F. Gerfers M. Ortmanns. *Continuous-Time Sigma-Delta A/D Conversion - Fundamentals, Performance Limits and Robust Implementation*. Springer, 2005.

[22] Kazuo Matsukawa et al. “A fifth-order continuous-time delta-sigma modulator with single-opamp resonator”. In: *Solid-State Circuits, IEEE Journal of* 45.4 (2010), pp. 697–706.

[23] “Optimal parameters for $\Delta\Sigma$ modulator topologies”. In: *IEEE Transactions on Circuits and Systems* (Sept. 1998).

[24] Maurits Ortmanns, Friedel Gerfers, and Yiannos Manoli. “Fundamental limits of jitter insensitivity in discrete and continuous-time sigma delta modulators”. In: *Circuits and Systems, 2003. ISCAS’03. Proceedings of the 2003 International Symposium on*. Vol. 1. IEEE. 2003, pp. I–I.

[25] Stephen H. Lewis Paul J. Hurst. “Determination of Stability Using Return Ratios in Balanced Fully Differential Feedback Circuits”. In: *IEEE Transactions on Circuits and Systems* (Dec. 1995).

[26] Shanthi Pavan and Prabu Sankar. “Power reduction in continuous-time delta-sigma modulators using the assisted opamp technique”. In: *Solid-State Circuits, IEEE Journal of* 45.7 (2010), pp. 1365–1379.

[27] Shanthi Pavan Pradeep Shettigar. “Design Techniques for Wideband Single-Bit Continuous-Time $\Delta\Sigma$ Modulators With FIR Feedback DACs”. In: *IEEE JOURNAL OF SOLID-STATE CIRCUITS* (2012).

[28] G.C. Temes R. Schreier. *Understanding Delta-Sigma Data Converters*. Wiley-Interscience, 2005.

[29] Dejan Radjen. “Continuous-Time $\Sigma\Delta$ Modulators for Ultra-Low-Power Radios”. PhD thesis. Lund University, Aug. 2014.

[30] Dejan Radjen et al. “A continuous time delta-sigma modulator with reduced clock jitter sensitivity through DSCR feedback”. In: *Analog Integrated Circuits and Signal Processing* 74.1 (2013), pp. 21–31.

[31] Dejan Radjen et al. “A low-power 2nd-order CT delta-sigma modulator with a single operational amplifier”. In: *Analog Integrated Circuits and Signal Processing* 80.3 (2014), pp. 387–397.

[32] Mohammad Ranjbar et al. “A 3.1 mW continuous-time $\Delta\Sigma$ modulator with 5-bit successive approximation quantizer for WCDMA”. In: *Solid-State Circuits, IEEE Journal of* 45.8 (2010), pp. 1479–1491.

[33] Richard Schreier. *Delta Sigma Toolbox*. Jan. 2000. URL: `http://www.mathworks.com/matlabcentral/fileexchange/19-delta-sigma-toolbox`.

[34] O. Shoaei. “Continuous-time delta-sigma A/D converters for high speed applications”. PhD thesis. Carleton University, 1995.

[35] Hai Tao, Laszlo Toth, and John M Khoury. “Analysis of timing jitter in bandpass sigma-delta modulators”. In: *Circuits and Systems II: Analog and Digital Signal Processing, IEEE Transactions on* 46.8 (1999), pp. 991–1001.

[36] Hung-Chieh Tsai et al. “A 64-fJ/Conv.-step continuous-time $\Sigma\Delta$ modulator in 40-nm CMOS using asynchronous SAR quantizer and digital $\Sigma\Delta$ truncator”. In: *Solid-State Circuits, IEEE Journal of* 48.11 (2013), pp. 2637–2648.

[37] Eric J vanderZwan and E Carel Dijkmans. “A 0.2-mW CMOS $\Sigma\Delta$ modulator for speech coding with 80 dB dynamic range”. In: *Solid-State Circuits, IEEE Journal of* 31.12 (1996), pp. 1873–1880.

[38] Chan-Hsiang Weng et al. “A 0.89-mW 1-MHz 62-dB SNDR continuous-time delta–sigma modulator with an asynchronous sequential quantizer and digital excess-loop-delay compensation”. In: *Circuits and Systems II: Express Briefs, IEEE Transactions on* 58.12 (2011), pp. 867–871.

[39] Jing Yang, Thura Lin Naing, and Robert W Brodersen. “A 1 GS/s 6 bit 6.7 mW successive approximation ADC using asynchronous processing”. In: *Solid-State Circuits, IEEE Journal of* 45.8 (2010), pp. 1469–1478.

[40] Ramin Zanbaghi and Terri S Fiez. “A novel low power hybrid loop filter for continuous-time sigma-delta modulators”. In: *Circuits and Systems, 2009. ISCAS 2009. IEEE International Symposium on.* IEEE. 2009, pp. 3114–3117.

[41] Ramin Zanbaghi, Pavan Kumar Hanumolu, and Terri S Fiez. “An 80-DB DR, 7.2-MHz bandwidth single opamp biquad based CT $\Delta\Sigma$ modulator dissipating 13.7-mW”. In: *Solid-State Circuits, IEEE Journal of* 48.2 (2013), pp. 487–501.