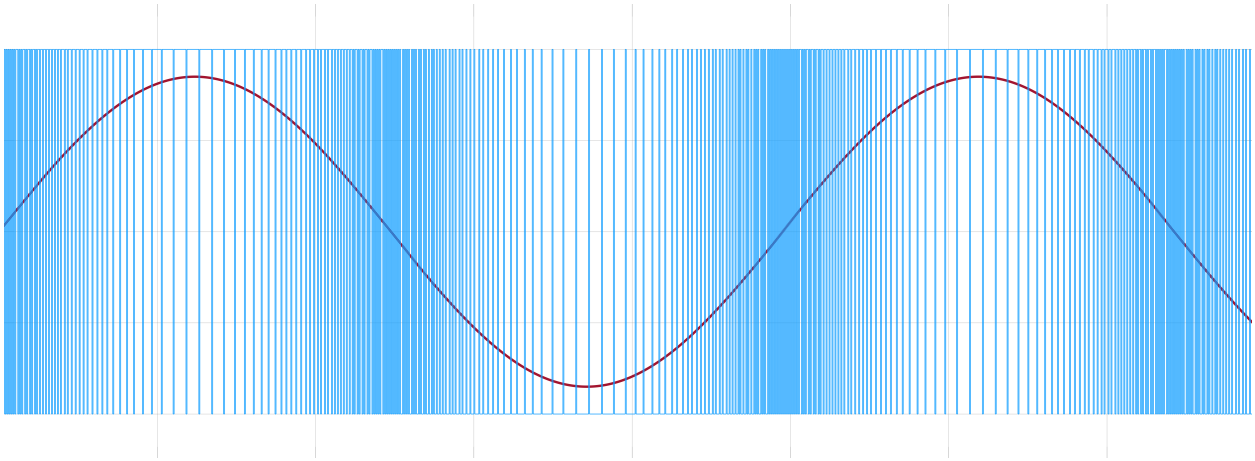

FPGA-based Noise Shaping ADC for Feedback in Switched Control Systems

Bachelor's Thesis for the Electrical Engineering Program TIELL

Nicklas Wright & Hampus Lang

DEPARTMENT OF ELECTRICAL ENGINEERING

CHALMERS UNIVERSITY OF TECHNOLOGY

Gothenburg, Sweden 2023

www.chalmers.se

BACHELOR'S THESIS 2023

FPGA-based Noise Shaping ADC for Feedback in Switched Control Systems

NICKLAS WRIGHT
HAMPUS LANG

CHALMERS

Department of Electrical Engineering
Chalmers University of Technology
Gothenburg 2023

FPGA-based noise shaping ADC for feedback in switched control systems
NICKLAS WRIGHT
HAMPUS LANG

Supervisor: Magnus Bik, Lab Gruppen
Examiner: Rob Maaskant, Department of Electrical Engineering

Bachelor's Thesis 2023
Department of Electrical Engineering
Chalmers University of Technology
SE-412 96 Gothenburg
Telefon +46 31 772 1000

Cover photo: Example photo of a $\Delta\Sigma$ modulated sine wave in Simulink.

Written in LaTeX
Gothenburg 2023

FPGA-based noise shaping ADC for feedback in switched control systems
NICKLAS WRIGHT
HAMPUS LANG
Department of Electrical Engineering
Chalmers University of Technology

Abstract

This thesis explores the possibility of implementing a customized Delta-Sigma ADC for feedback in a digital, switched audio amplifier. It was conducted in collaboration with *Lab Gruppen*, Kungsbacka, who wished to investigate if a cheaper and more flexible ADC than the current ones available on the market could be designed, using an FPGA and simple analog circuitry. The design process involved system-level design in Matlab/Simulink, analog circuit design in LTspice and digital hardware implementation in Modelsim and Vivado. Simulation results and calculations showed that a 1-bit, second order Delta-Sigma modulator followed by a third order CIC-decimation filter could, in theory, fulfill the product specifications. An attempt was made to implement the analog hardware using discrete components, and was achieved for a first order modulator in simulation. The digital CIC-filter was successfully designed, synthesized and implemented on the target device. Although the proposed design was not successfully implemented in hardware, the results show promise in finding an implementation that is cheaper than off-the-shelf ADCs. Future work could include designing the second order modulator using more conventional switched-capacitor integrators, improving the digital filter and building a functioning prototype.

Keywords: Delta-Sigma, ADC, oversampling, first order, second order, 1-bit, FPGA, noise shaping

Acknowledgements

This thesis was an arduous and complex project to plan and perform from start to finish, and could not have been done without the help of two people that deserve our gratitude.

Firstly, we would like to thank our examiner, Rob Maaskant, who provided important guidance in planning the project as well as in writing this report.

Secondly, we want to especially thank our technical supervisor at Lab Gruppen, Magnus Bik, who formulated the idea for the project and whose master's thesis this work was based on. Magnus was always available to share his great insight in high-speed signal processing and gave us crucial feedback and help during discussions.

Nicklas Wright & Hampus Lang, Gothenburg, June 2023

List of Abbreviations

Below is the list of acronyms that have been used throughout this thesis listed in alphabetical order:

ADC	Analog-To-Digital Converter
CIC	Cascaded Integrator-Comb (filter)
CT	Continuous Time
DAC	Digital-To-Analog Converter
dBFS	Decibels (relative to) Full-Scale
DFF	D flip-flop
DT	Discrete Time
ESL	Equivalent Series Inductance
ESR	Equivalent Series Resistance
FPGA	Field-Programmable Gate Array
GBP	Gain-Bandwidth Product
HP	High-Pass
KCL	Kirchhoff's Current Law
LP	Low-Pass
MOSFET	Metal-Oxide Semiconductor Field-Effect Transistor
NMOS	N-Channel MOSFET
NTF	Noise Transfer Function
OP-AMP	Operational Amplifier
OSR	Oversampling Ratio
PCM	Pulse Code Modulation
PWM	Pulse Width Modulation
RCA	Ripple-Carry Adder
SINAD	Signal-To-Noise and Distortion Ratio
SMD	Surface Mount Device
SNR	Signal-To-Noise Ratio
SQNR	Signal-To-Quantization-Noise Ratio
SR	Slew-Rate
STF	Signal Transfer Function
THD	Total Harmonic Distortion
VHDL	Very High-Speed Integrated Circuit Hardware Description Language

List of Symbols

Below is the nomenclature of the parameters and variables that have been frequently used throughout this thesis.

N	Differential delay
R	Downsampling factor
f_{B}	Bandwidth upper-limit frequency
f_{N}	Nyquist frequency
f_{s}	Sampling frequency
Δ	Quantization step
ϵ_{Q}	Quantization error
p	Probability density function
μ	Arithmetic mean value
σ^2	Variance
τ	RC time constant

Contents

List of Abbreviations **ix**

List of Symbols **xi**

List of Figures **xv**

List of Tables **xvii**

1 Introduction **1**
1.1 Background . . . 1
1.2 Aim . . . 2
1.3 Delimitations . . . 2
1.4 Problem Statement . . . 2
1.5 Thesis Outline . . . 3

2 Theoretical Background **5**
2.1 The $\Delta\Sigma$ Architecture . . . 5
2.2 Quantization and Quantization Noise . . . 6
2.3 Noise Shaping and Oversampling . . . 8
2.4 Second Order Modulator . . . 10
2.5 Signal-to-Noise Ratio . . . 11
2.6 Analog Electronics . . . 13
2.6.1 Sample-and-hold . . . 13
2.6.2 Integrator . . . 15
2.6.3 Comparator . . . 20
2.6.4 Slew-rate . . . 20
2.7 Digital Decimation Filter . . . 21
2.8 I2S Serial Interface . . . 22

3 Design and Implementation **23**
3.1 Simulink Simulations . . . 23
3.2 Analog Circuit Design . . . 28
3.2.1 S/H circuit . . . 28
3.2.2 First order circuit . . . 30
3.2.3 Second order circuit . . . 31
3.2.4 Hardware implementation . . . 32
3.3 Digital Hardware Design . . . 33

3.3.1 Clock generation 33
3.3.2 Decimation filter 34
3.3.3 Serial communication 35

4 Results **37**
4.1 Circuit Simulations 37
4.2 CIC-filter 40

5 Conclusions and Recommendations **43**

Bibliography **45**

A Ideal Op-Amp Schematic **I**

B Top Level VHDL-code **III**

List of Figures

2.1 *First order, n-bit modulator block diagram.* 5
2.2 *Conceptual block-diagram of a first order, n-bit ΔΣ ADC.* 6
2.3 *Linear model of first order ΔΣ modulator.* 8
2.4 *Amplitude response of the first order NTF.* 9
2.5 *Linear model of second order ΔΣ modulator.* 10
2.6 *Comparison of the amplitude response of a first and second order NTF.* 11
2.7 *Sample-and-Hold circuit.* 13
2.8 *Ideal integrator circuit.* 15
2.9 *Practical integrator circuit.* 16
2.10 *Frequency response of the ideal and practical integrator.* 18
2.11 *Practical summing integrator circuit.* 19
2.12 *Example comparator circuit.* 20
2.13 *First order CIC decimation filter.* 21

3.1 *First order modulator.* 23
3.2 *Second order modulator.* 24
3.3 *Periodogram of first and second order 1-bit ΔΣ with OSR = 256.* . . . 25
3.4 *Integrator output of the 1-bit ΔΣ with OSR = 256.* 26
3.5 *3-bit quantizer output signal.* 27
3.6 *Sample-and-hold circuit.* 28
3.7 *S/H output with OSR = 128.* 29
3.8 *First order circuit.* 30
3.9 *Second order circuit.* 32
3.10 *Hardware realization of decimation filter.* 34
3.11 *Hardware implementation of an n-bit shift register.* 35

4.1 *Simulated results of the sampled voltage and the filtered output.* 37
4.2 *Integrator output.* 38
4.3 *Power spectrum of first order circuit with OSR = 256 after 100 periods.* 39
4.4 *Simulink model of CIC decimation filter.* 40
4.5 *Power spectrum of digital filter output.* 41
4.6 *Sine wave from LUT amd digital filter output.* 41

A.1 *Ideal op-amp modeled as a voltage dependant voltage source.* I

List of Tables

3.1 *SNR simulation result for first and second order* $\Delta\Sigma$. 24
3.2 *Theoretical SQNR for first and second order* $\Delta\Sigma$. 24
3.3 *SNR simulation result for 3-bit first order* $\Delta\Sigma$. 27
3.4 *Clock table.* 33

1
Introduction

Switched audio amplifiers require an analog audio signal as input to operate. Therefore, they have traditionally been designed and built completely in the analog domain using analog feedback. Today most audio systems use digital interfacing and audio processing, rendering a complete analog system undesirable and a thing of the past. Because of this significantly digital architecture, it would be logical to build amplifiers in the digital domain to avoid unnecessary signal conversions. However, the voltage waveform across the speaker driver must be analog, regardless of if the power amplifier is analog or digital. To feed this signal back into the digital control loop, analog-to-digital conversion is therefore required.

When using digital feedback, it is of the essence to achieve a high gain margin and low output noise. This requires a low latency measurement of the control node. Additionally, it is crucial that the sampling frequency of the signal is high enough to avoid aliasing of the signal – otherwise the converted digital signal would not contain the same information as the analog audio signal.

1.1 Background

This degree project is a continuation of a master's thesis [1] conducted at the audio technology company *Lab Gruppen* located in Kungsbacka, Sweden, in which a design of the system just described was implemented. During that work it was discovered that the analog-to-digital converters (ADCs) available on the market with the high sampling rate and precision demanded of the digital feedback unfortunately are quite expensive for the digital architecture proposed to be financially viable for large scale manufacturing. Rather than abandoning the design completely, Lab Gruppen wishes to explore if it is possible to reduce the cost of the sampling system by implementing a simpler design using discrete components.

A popular choice of converter for this task is a so-called Delta-Sigma ($\Delta\Sigma$) ADC, due to its ability of achieving high resolution and low noise with simple circuitry, by utilizing oversampling and noise shaping. Commonly, the implementation of such a high-speed sampling and conversion system involves a Switched-Capacitor (SC)–integrator which effectively achieves sampling and integration of the signal by using only one op-amp [2, p. 288]. This design, while effective, is complex and often requires intricate differential and compensation circuits to get to operate properly.

This project instead investigates the possibility of implementing a functioning $\Delta\Sigma$ ADC using a modular and simpler approach for the design.

1.2 Aim

This thesis aims to investigate the possibility of implementing a custom-made sampling system for a specific amplifier design in order to improve noise performance and reduce cost in relation to using an off-the-shelf ADC. The project is research based, meaning that a finished product is not necessarily the end-goal. Arriving at a conclusion that a cheaper alternative to the products available on the market is possible or not, using the suggested design, is a valid result either way.

A complete model of the design that satisfies the system specifications should be created and tested in Simulink. This design should then, ideally, be implemented in hardware with the help of an FPGA, using as few analog components as possible in order to keep the design cheap and flexible. A performance evaluation of the system should be conducted, and the results compared to that of the simulation model. Possible areas of improvement that are discovered should be addressed. Again, this depends on whether a working prototype is designed or not.

1.3 Delimitations

In order to sample the output signal from the power stage of the amplifier, the signal must be scaled in order to fit the maximum input voltage swing of the ADC. This requires an analog voltage divider circuit. However, the input signal to the ADC will be assumed to be properly scaled, and this circuit will therefore not be considered.

The project aims to investigate the possibility of implementing as much of the design as possible in the digital domain. This includes the anti-aliasing filter. However, should it be concluded that the filter is best implemented in analog hardware, this circuit will be omitted in the final design since the design of analog filters is already quite thoroughly researched.

A digital filter will be designed in order to test the complete ADC-system. However, this filter will not be optimized since the subject of digital filter design is heavily researched already.

1.4 Problem Statement

The specifications of the sampling system are as follows:

- Aliasing-free below a frequency of 2 MHz
- At least 20 kHz bandwidth
- At least 100 dB SNR for frequencies 0 – 20 kHz

- A maximum of -80 dB THD for frequencies 0 – 20 kHz

The above-listed requirements should be achieved using a sampling system consisting of a Delta-Sigma ($\Delta\Sigma$) ADC. This ADC should be used to sample the output node of a switched audio amplifier. The amplifier in question generates a 3-level PWM (Pulse Width Modulated) signal with a switching frequency of 2.016 MHz.

1.5 Thesis Outline

Sec. 1.1 – 1.4 of this chapter provides an overview of why this project was done, what boundaries and limitations were set beforehand, as well as what was hoped to be achieved at the end. In Ch. 2, an in-depth overview of the theory used in the design process is presented. Ch. 3 details the method used when designing and implementing the system and Ch. 4 presents the final results of the finished design. Finally, Ch. 5 conclusions are drawn about the performance of the implemented system, as well as providing recommendations for future improvements.

2
Theoretical Background

This chapter presents relevant theory supporting the design choices made during the process of designing the complete system.

2.1 The $\Delta\Sigma$ Architecture

The Delta-Sigma ($\Delta\Sigma$) modulator gets its name from the structure of its block-diagram. Displayed in Fig. 2.1 is an n-bit, first order $\Delta\Sigma$ modulator.

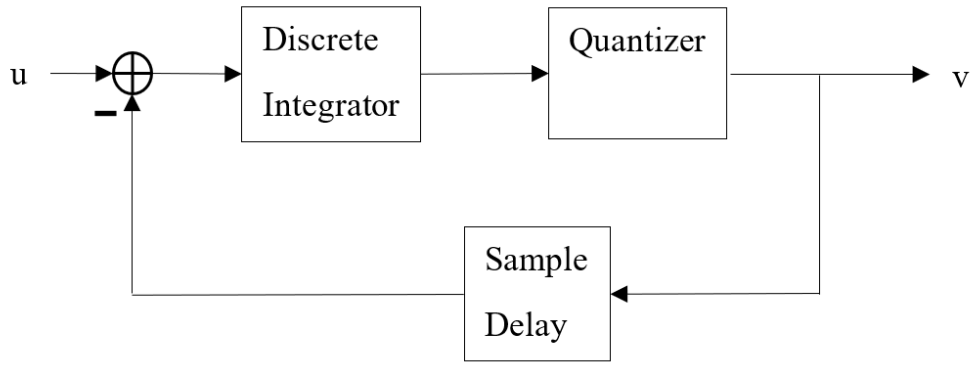

Figure 2.1: *First order, n-bit modulator block diagram.*

As depicted in Fig. 2.1, the sum (sigma) refers to the integrator and the difference (delta) to the subtraction of the output from the input. Consequently, the integrator output becomes the sum of the difference. The order of the mathematical operations results in the name "Delta-Sigma" ($\Delta\Sigma$). However, this is a matter of preference since "Sigma-Delta" is commonly used equivalently [3, p. 46].

The integrator block in Fig. 2.1 is implemented using a discrete transfer function (Z-transform), indicating that the system is sample-based. Since there cannot exist a delay-free loop, the feedback is delayed with one sample period. The quantizer block is realized using a cascaded ADC-DAC structure, where the output of the ADC is connected to the output of the modulator and fed back to the modulator input via a DAC.

What the system in Fig. 2.1 achieves is an accumulation of the difference between the input signal and its digital approximation. That is, the output of the modulator is not a direct representation of the input signal [3, p. 8], but rather a pulse code modulated (PCM) bit stream of logic ones and zeros. The input can only be approximated through a large number of iterations (samples) averaged on the output, granted that the value of the input signal effectively does not change between successive samples. This leads to the fact that $\Delta\Sigma$ modulation relies on the input being heavily oversampled (more in Sec. 2.2). Intuitively, increasing the number of samples being accumulated and averaged improves the resolution of the modulator.

The result of embedding the quantizer in a feedback-loop and oversampling is that the ADC and DAC used in the loop can be very coarse, while the modulator still achieves high resolution after averaging its output [3, p. 9]. Even a 1-bit quantizer, which can be realized by a simple comparator, can achieve very good noise-performance. The combination of the $\Delta\Sigma$ modulator (shown in Fig. 2.1) and a moving average filter on its output is referred to as first order $\Delta\Sigma$ ADC [3, p. 10]. Fig. 2.2 shows a conceptual block-diagram of such an ADC.

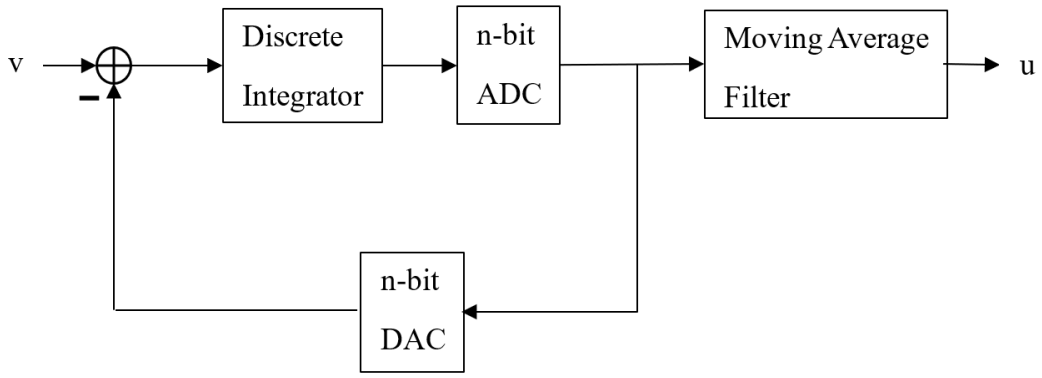

Figure 2.2: *Conceptual block-diagram of a first order, n-bit $\Delta\Sigma$ ADC.*

The block diagram in Fig. 2.2 takes a sampled input signal and outputs a digital approximation of the input. For example, if the input to the ADC is larger than some threshold value it outputs a logic '1', and similarly a logic '0' if it is lower. The moving average filter has several functions and is realized in the digital domain (discussed in more detail in Sec. 2.7). It is again essential that the input is heavily oversampled for the system to operate correctly.

2.2 Quantization and Quantization Noise

The quantization discussed in the previous section refers to changing the continuous amplitude of the sampled input signal to discrete. This is done by dividing the dynamic range of the system into equally sized *quantization intervals* [2, p. 15]. If the minimum detectable amplitude of the quantizer is denoted by X_{min}, the maximum amplitude by X_{max}, and the number of quantization intervals by M, then the

quantization step Δ is defined as

$$\Delta = \frac{X_{\max} - X_{\min}}{M} \tag{2.1}$$

The discrete value assigned by a quantizer to a continuous input depends on which quantization interval it lies in. In most cases, the value assigned to the input is the mid-point of the interval. This introduces a *quantization error* since the input signal can take any value in an interval but is always assigned the mid-point value [2, p. 15]. The output of the quantizer can be described by

$$Y = X_{\text{in}} + \epsilon_{\text{Q}} = (n + 1/2)\Delta, \quad n\Delta < X_{\text{in}} < (n + 1)\Delta \tag{2.2}$$

where X_{in} is the quantizer input, ϵ_{Q} is the quantization error and n is the interval where X_{in} resides. It follows from Eq. (2.2) that $\epsilon_{\text{Q}} \in [-\Delta/2, +\Delta/2]$. However, ϵ_{Q} is a strongly non-linear function of the input X_{in} [3, p. 31]. To simplify mathematical analysis of the quantization error, it is assumed that the error is an additive "noise" sequence independent of the input. This noise is also assumed to be a white sequence, uniformly distributed in $[-\Delta/2, +\Delta/2]$ [3, p. 32]. Thus, the quantization error is referred to as *quantization noise*.

The probability density function of the uniformly distributed quantization noise can be described by [2, p. 18]

$$p(\epsilon_{\text{Q}}) = \begin{cases} 1/\Delta, & \epsilon_{\text{Q}} \in [-\Delta/2, +\Delta/2] \\ 0, & \text{otherwise} \end{cases} \tag{2.3}$$

From Eq. (2.3) the arithmetic mean value μ and the variance σ^2 can be calculated as [4, p. 498]

$$\mu = \frac{-\Delta/2 + \Delta/2}{2} = 0 \tag{2.4}$$

$$\sigma^2 = \frac{(-\Delta/2 + \Delta/2)^2}{12} = \frac{\Delta^2}{12} \tag{2.5}$$

Since the mean value (DC-value) is 0, and the unit is volts, the variance of the quantization noise is equal to the noise power [4, p. 496]. From Eq. (2.4) and Eq. (2.5) it is clear that the noise power decreases with the quantization step. Furthermore, if each mid-point value of the quantizer is represented by a digital code then decreasing the quantization step is equivalent to increasing the number of bits.

In reality the quantization error is not white noise, especially when the number of quantization levels are few [2, p. 17]. This is the case for most $\Delta\Sigma$ converters, where noise shaping and oversampling compensate for the low number of bits and thus eliminating some, or all, non-linearity caused by the multi-level feedback DAC, which otherwise would require complex circuitry to mitigate [2, p. 264]. The quantization error forms a repetitive pattern of logic ones and zeros that give rise to spectral tones, referred to as *pattern noise* [2, p. 294]. Hence, the quantizer also produces harmonics; effectively increasing the total harmonic distortion (THD) and concentrating the power of the quantization error at certain frequencies instead of

spreading it over the entire signal bandwidth [2, p. 295].

These harmonics can be suppressed by a technique called *dithering*, where high-frequency and low amplitude "noise", for example a sine wave, is added to the loop which introduces a chaotic element that breaks up the periodic pattern of the quantizer error [2, p. 295]. This "noise" must be low in amplitude as to not affect the input signal too much, and high enough in frequency so it can be filtered out by the digital filter at the output of the modulator. Practically, thermal noise from analog components can provide this functionality.

In conclusion, the quantization error (or noise) is an unavoidable consequence of data conversion. It can only be fully eliminated when the number of bits is increased to infinity, which is not possible in practice [2, p. 16].

2.3 Noise Shaping and Oversampling

Noise shaping and *oversampling* are two fundamental methods employed in the $\Delta\Sigma$ modulator in order to reduce the in-band quantization noise. When analyzing the first order $\Delta\Sigma$ modulator as a linear system, the quantizer can be modeled as an added noise source e, as depicted in Fig. 2.3.

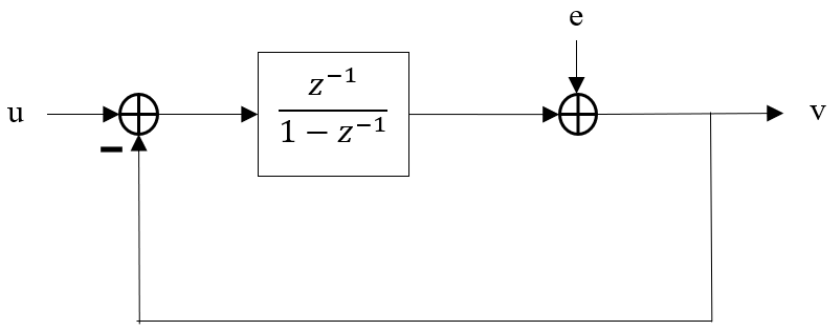

Figure 2.3: *Linear model of first order $\Delta\Sigma$ modulator.*

In said figure, the sample delay has been pushed into the forward path, resulting in an equation describing the relation between input and output:

$$[U(z) - V(z)]\frac{z^{-1}}{1 - z^{-1}} = V(z) \Leftrightarrow V(z) = U(z) \cdot z^{-1}. \tag{2.6}$$

The equation relating the noise source to the output becomes:

$$-V(z)\frac{z^{-1}}{1 - z^{-1}} + E(z) = V(z) \Leftrightarrow V(z) = E(z) \cdot (1 - z^{-1}). \tag{2.7}$$

Using (2.6) and (2.7), the total system can be described by

$$V(z) = U(z) \cdot STF(z) + E(z) \cdot NTF(z) = U(z) \cdot z^{-1} + E(z) \cdot (1 - z^{-1}), \tag{2.8}$$

where $STF(z)$ is the signal transfer function and $NTF(z)$ the noise transfer function [2, p. 258]. As can be seen from Eq. (2.8), the input is the delayed by one clock cycle while the quantization noise passes through $(1 - z^{-1})$. Analyzing $NTF(z)$ in the frequency domain by replacing z with $e^{j\omega T}$, where ωT is the normalized frequency, yields [2, p. 258]

$$NTF(e^{j\omega T}) = 1 - e^{j\omega T} = 2j \cdot e^{-j\omega T/2} \cdot \frac{e^{j\omega T/2} - e^{j\omega T/2}}{2j} = 2je^{-j\omega T/2} \sin(\omega T/2). \quad (2.9)$$

Then from Eq. (2.9) the squared amplitude can be calculated as

$$|NTF(e^{j\omega T})|^2 = 4 \sin^2(\omega T/2). \quad (2.10)$$

Eq. (2.10) shows that the quantization noise power, although amplified by a factor four, is shaped by $\sin^2(\omega T/2)$. That is, the in-band quantization noise is high-pass (HP) filtered by the modulator. Fig. 2.4 shows the amplitude response of the NTF for frequencies up to the sampling frequency ($\omega = \pi$).

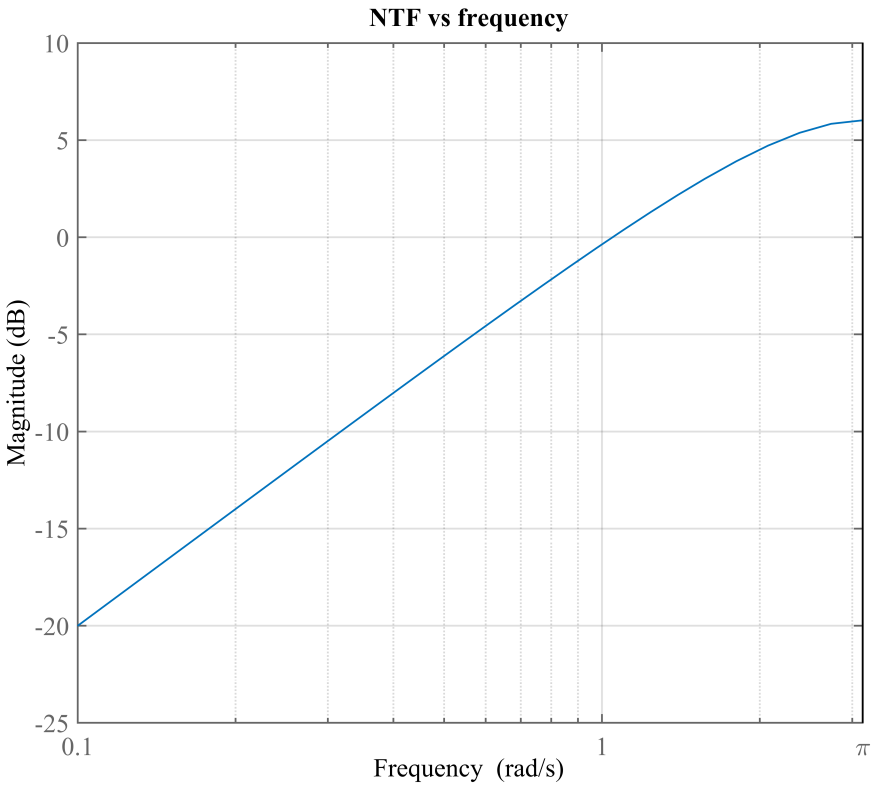

Figure 2.4: *Amplitude response of the first order NTF.*

The plot in Fig. 2.4 shows significant attenuation of lower frequencies in the signal band. An increase in magnitude of 20 dB/decade can also be seen. To further exploit the fact that lower frequencies are subjected to more attenuation, oversampling is introduced. Oversampling is measured using the *oversampling ratio* (OSR), defined as [3, p. 11]

$$OSR = f_s/2f_B, \quad (2.11)$$

where f_s is the sampling frequency and f_B is the signal bandwidth. The OSR specifies how much faster than the *Nyquist rate* of two times the bandwidth the system is sampled. Since the quantization noise power is distributed evenly from 0 to f_s [4, p. 515], a consequence of oversampling is that the in-band noise floor is lowered. This happens because the signal band still only occupies 0 to f_B, while the quantization noise is spread evenly over 0 to OSR $\cdot\, 2f_B$.

The in-band (unshaped) quantization noise power can now be described by [2, p. 253]

$$V_{n,B}^2 = \frac{\Delta^2}{12} \cdot \frac{2f_B}{f_s} = \frac{V_{ref}^2}{12 \cdot 2^{2n}} \cdot \frac{1}{OSR}, \tag{2.12}$$

which applies for a quantizer with n bits, 2^n levels and a dynamic range of V_{ref}. From Eq. (2.12) it can be seen that the noise power, as expected, decreases as OSR increases.

2.4 Second Order Modulator

As discussed previously, the performance of the first order $\Delta\Sigma$ modulator can be improved by using a quantizer with more levels, or by increasing the OSR. Another possibility for improving noise performance is choosing a quantizer with small in-band quantization noise spectral density [3, p. 63]. That is, instead of decreasing the noise spectral density for all frequencies by increasing the number of levels, reduce the noise only in the band of interest. A quantizer with exactly those characteristics, as described in the previous section, is the $\Delta\Sigma$ modulator.

By replacing the quantizer in Fig. 2.1 with another $\Delta\Sigma$ modulator and doing some rearranging, the linear model of the second order modulator is obtained in Fig. 2.5.

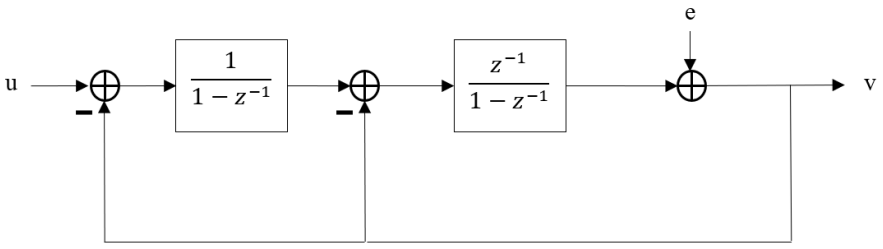

Figure 2.5: *Linear model of second order $\Delta\Sigma$ modulator.*

Analyzing the system in Fig. 2.5 yields

$$V(z) = U(z) \cdot STF(z) + E(z) \cdot NTF(z) = U(z) \cdot z^{-1} + E(z) \cdot (1 - z^{-1})^2. \tag{2.13}$$

Comparing this to Eq. (2.8), the noise is now passed through $(1 - z^{-1})^2$ instead of $(1 - z^{-1})$. This results in better noise shaping as can be seen from the frequency domain function of the NTF [2, p. 267]:

$$NTF(e^{j\omega T}) = (1 - e^{-j\omega T})^2 = -4e^{-j\omega T} \sin^2(\omega T/2). \tag{2.14}$$

The resulting amplitude response plot is displayed in Fig. 2.6.

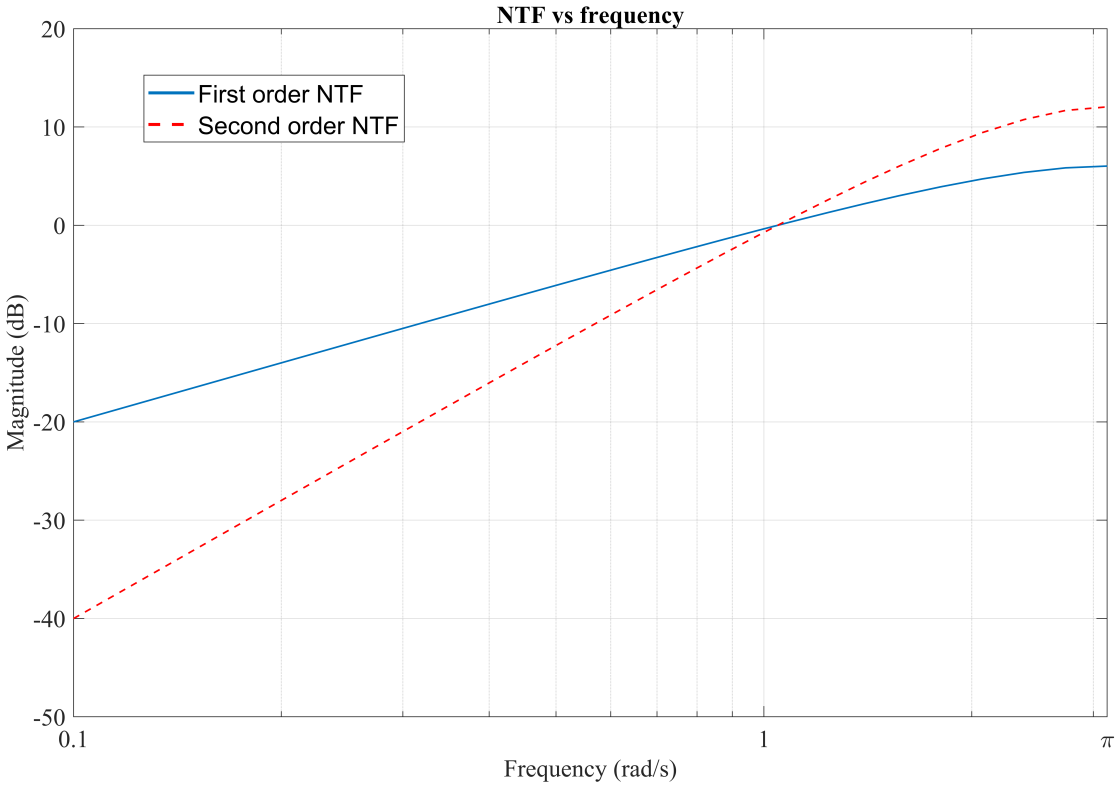

Figure 2.6: *Comparison of the amplitude response of a first and second order NTF.*

As Fig. 2.6 shows, the second order modulator achieves better noise suppression at low frequencies compared to the first order modulator, as well as a 40 dB/decade increase in magnitude up to the sampling frequency.

2.5 Signal-to-Noise Ratio

The signal-to-noise ratio (SNR) is defined as the power of a signal passed through a circuit divided by the power of the noise generated by the circuit across the entire Nyquist-band [2, p. 62]. The following derivations were based on [2, p. 259].

The in-band square noise voltage of the shaped spectrum of the first order $\Delta\Sigma$ modulator can be described by

$$V_{\mathrm{n}}^2 = v_{\mathrm{n,Q}}^2 \int_0^{f_{\mathrm{B}}} |NTF(f)|^2 \,\mathrm{d}f = v_{\mathrm{n,Q}}^2 \int_0^{f_{\mathrm{B}}} 4\sin^2(\pi f T) \,\mathrm{d}f \approx v_{\mathrm{n,Q}}^2 \frac{4\pi^3}{3} f_{\mathrm{B}}^3 T^2 \tag{2.15}$$

with the approximation $\sin(x) \approx x$ which applies for $\omega_{\mathrm{B}} \frac{T}{2} \ll \frac{\pi}{2}$, i.e., when the sampling frequency is much larger than the bandwidth. Then since the square noise voltage of the unshaped spectrum across the bandwidth is

$$V_{\mathrm{n,Q}}^2 = \int_0^{f_{\mathrm{s}}/2} v_{\mathrm{n,Q}}^2 \,\mathrm{d}f = v_{\mathrm{n,Q}}^2 \frac{f_{\mathrm{s}}}{2} \tag{2.16}$$

and $T = 1/f_s$, Eq. (2.15) can be rewritten as

$$V_n^2 = \frac{2V_{n,Q}^2}{f_s}\frac{4\pi}{3}f_B^3\left(\frac{1}{f_s}\right)^2 = V_{n,Q}^2\frac{\pi^2}{3}\left(\frac{f_B}{f_s/2}\right)^3 = V_{n,Q}^2\frac{\pi^2}{3}OSR^{-3} \tag{2.17}$$

From Eq. (2.1) it is known that a quantizer with range $0 - V_{ref}$ and k quantization thresholds have a quantization step of $\Delta = V_{ref}/k$. Then from Eq. (2.5), the power of the unshaped quantization noise can be written as

$$V_{n,Q}^2 = \frac{V_{ref}^2}{k^2 \cdot 12}. \tag{2.18}$$

Moreover, the power of a full-scale sine wave is

$$V_{sine}^2 = \frac{V_{ref}^2}{8}. \tag{2.19}$$

Using Eq. (2.17), (2.18) and (2.19) the SNR of the first order $\Delta\Sigma$ modulator can be determined by

$$SNR_{\Delta\Sigma 1} = \frac{V_{sine}^2}{V_n^2} = \frac{12}{8} \cdot k^2 \cdot \frac{3}{\pi^2} \cdot OSR^3, \tag{2.20}$$

which in decibels equates to

$$SNR_{\Delta\Sigma 1,dB} = 6.02 \cdot \log_2(k) + 1.78 - 5.17 + 9.03 \cdot \log_2(OSR). \tag{2.21}$$

Eq. (2.21) shows that doubling the OSR results in a 9.03 dB increase in SNR. It can also be seen that doubling the number of quantization intervals secures a 6.02 dB improvement of the SNR.

Performing the same analysis as described above on the second order modulator yields [2, p. 267]

$$SNR_{\Delta\Sigma 2,dB} = 6.02 \cdot \log_2(k) + 1.78 - 12.9 + 15.05 \cdot \log_2(OSR). \tag{2.22}$$

The same improvement of the SNR as for the first order modulator can be seen for the second order modulator when increasing the quantizer resolution. However, when doubling the OSR an SNR increase of 15.05 dB is now achieved.

It is worth mentioning that these formulas for the SNR only apply if one assumes the quantization error to be purely additive white noise, as described in Sec. 2.2. It is also lacking the additional noise that will be added from the active and passive components in the analog circuit. Furthermore, it does not include the in-band harmonics created by the quantizer, which was also discussed in Sec. 2.2. This is technically called signal-to-noise and distortion ratio (SINAD). However, SNR is used equivalently in this report. To distinguish the noise analysis between the linearized and real-world model, signal-to-quantization-noise ratio (SQNR) will be from now on be used to describe the former, and SNR to describe the latter.

2.6 Analog Electronics

To realize the $\Delta\Sigma$ converter, some analog circuitry is needed. The fundamental parts of this circuit are threefold, consisting of: a sampling circuit, an integrator circuit and a quantizer circuit. The design of these varies, depending on which type of converter is to be designed. To avoid needlessly superfluous explanations, this section will only discuss the analog circuitry actually used in the project.

2.6.1 Sample-and-hold

In this project a discrete-time (DT) $\Delta\Sigma$ ADC was designed, meaning that the sampling of the input signal occurs before the modulator - as opposed to a continuous-time (CT) $\Delta\Sigma$ where the sampling is done inside the feedback loop [2, p. 325]. A CT modulator would omit the need for an external sampling circuit, since the sampling would be done inside of the FPGA. However, CT modulators require frequency-scaling of the integrators. This means that a system sampled with a frequency of m Hz requires integrators with a DC-gain of m [3, p. 229]. This would impose problems when designing the analog circuitry since a sampling frequency of 10.24 MHz will be used, resulting in a DC-gain of approximately 140 dB which is hard to find in op-amps. Furthermore, designing a linearized model of the CT modulator is more complex compared to DT. This is the case since the impulse response of the DAC needs to be carefully modeled. Using continuous transfer functions in Simulink also prohibits the use of fixed time-step simulation, which leads to unreasonably long simulation times.

In order for the AD-conversion to work, it is crucial to ensure that the system can accurately read exactly one value, or sample, of the continuous input signal and hold it until the quantization step is complete and the ADC is ready for its next iteration. One simple way of achieving this is with a so-called *Sample-and-Hold* (S/H) circuit. An implementation of this type of circuit can be seen in Fig. 2.7.

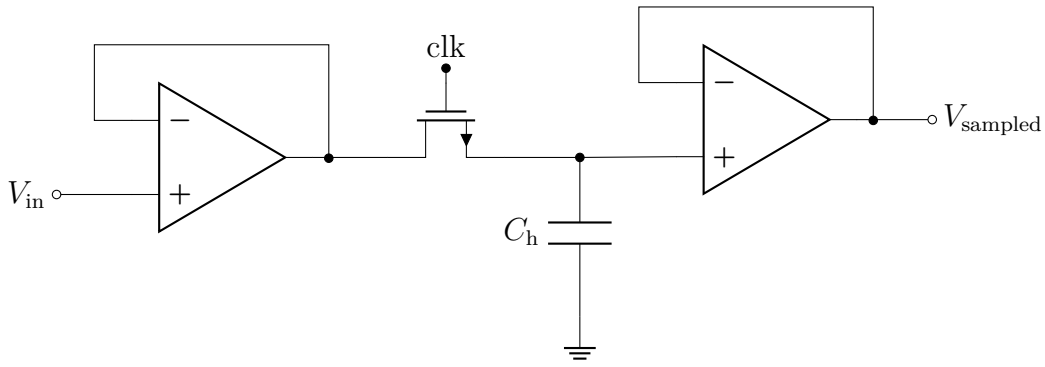

Figure 2.7: *Sample-and-Hold circuit.*

A switch, in this case realized by an NMOS-transistor, is toggled on and off by an external clock signal (clk) with a frequency equal to the sampling frequency f_{s}. When

the switch is on (t_{on}) the transistor starts conducting and charges the capacitor C_{h} to the same level as the input voltage. When the switch is off (t_{off}), the transistor acts as an open circuit and therefore no current flows through it. The voltage-stiff characteristics of the capacitor allows it to retain the sampled voltage for a short period of time [2, p. 210]. In other words, the capacitor samples the input voltage during t_{on}, and holds it during t_{off}.

The capacitance of C_{h} can be calculated using the time constant $\tau = RC$, where in this case R is the equivalent series resistance (ESR) of the circuit, which is the non-ideal capacitor's internal resistance as well as the conduction resistance ($R_{\text{DS(on)}}$) of the MOSFET. The time constant indicates the time required for the capacitor to reach its new steady state [6, p. 369], which by definition is the time it takes for a capacitor to charge to approximately 63 % of its input voltage when connected to a DC-source, or discharge to approximately 37 % of its initial charge value.

Modelling the capacitor with $R = \text{ESR}$ as a simple RC-circuit gives, in the frequency domain, the transfer function

$$H(j\omega) = \frac{1}{1 + j\omega RC_{\text{h}}}, \tag{2.23}$$

which is a low-pass (LP) filter with a cutoff frequency of $\omega_c = \frac{1}{RC_{\text{h}}}$ [7, p. 228]. In other words, the cutoff frequency is related to the time constant τ by

$$\omega_c = 2\pi f_c = \frac{1}{RC_{\text{h}}} \Leftrightarrow \tau = \frac{1}{2\pi f_{\text{c}}} = RC_{\text{h}} \tag{2.24}$$

In order for the sampling to work as intended the capacitor needs to reach its steady state during t_{on}, so $\tau \leq t_{\text{on}}$. This means that the capacitor needs to be fast enough to ensure that it can charge up to V_{in} every $t_{\text{on}} = T_{\text{s}}/2$. Setting the cutoff frequency equal to $2f_{\text{s}}$ ensures this. Given this fact, and by rewriting Eq. (2.24), the time constant RC_{h} can be calculated. The actual capacitance needed can then be determined by assuming some of the highest possible values of the equivalent series resistances, which typically are very low. This ensures that the minimal capacitance required for the sampling time is selected. The fact that the actual resistance most likely is lower than the assumed only serves to lower the sampling time even further; effectively making the charging time of the capacitor faster than necessary.

The high sampling frequency needed for the $\Delta\Sigma$ converter leads to the conclusion that a very small capacitance is needed. Furthermore, in this frequency range, the equivalent series inductance (ESL) of the capacitor starts being an issue as well - giving the impedance more inductive characteristics when $f_{\text{s}} \to \infty$. To mitigate this effect, a ceramic capacitor is preferable to use as C_{h} since they are internally not constructed as a coil and therefore have a low inductance [8, p. 335].

The two op-amps in Fig. 2.7 are configured as simple voltage followers, acting as buffers on the input and output of the S/H circuit. The input buffer is necessary to reduce the input load, since the S/H circuit's low impedance otherwise would

cause high currents in the circuit, resulting in high power loss which would affect the accuracy of the sampling [2, p. 210]. Furthermore, the output buffer ensures that the voltage is relayed to the output without discharging the capacitor [8, p. 348].

2.6.2 Integrator

As previously mentioned, summing the differences of input and the quantized output in discrete-time requires an integrator. There is another reason why an integrator is necessary. Replacing the integrator block in Figure 2.3 with a gain block with gain A, gives the following equation:

$$\begin{aligned} V(z) &= \left(\frac{A}{1+Az^{-1}}\right)U(z) + \left(\frac{1}{1+Az^{-1}}\right)E(z) \quad = \\ &= STF(z)U(z) + NTF(z)E(z). \end{aligned} \tag{2.25}$$

From Eq. (2.25) it is clear that when $A \to \infty$ then the $STF \to 1$ and the $NTF \to 0$. Hence, it would at first glance seem reasonable to have an infinite frequency-independent gain in the system to remove all quantization noise [3, p. 42]. However, determining the root of the polynomial in the two transfer functions' common denominator polynomial gives a single pole at $z = -A$. For a system to be stable, its poles need to be within the unit circle in the Z-domain, in other words, $|A| \leq 1$. Thus, making A too large over all frequencies would result in an unstable system [3, p. 44]. The best solution is to make A as high as possible only over the lower frequencies, and a simple component that achieves this is an integrator [3, p. 44].

An ideal integrator circuit can be constructed with an op-amp with a capacitor connected in negative feedback from its output. This circuit can be seen in Fig. 2.8.

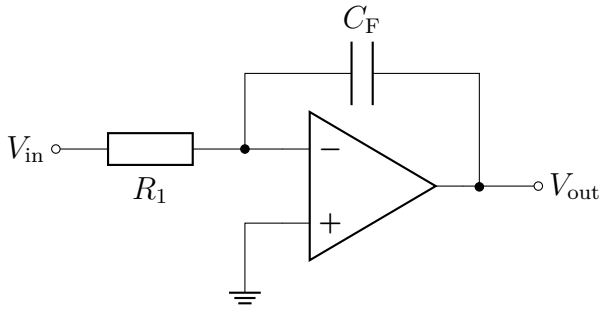

Figure 2.8: *Ideal integrator circuit.*

A function of the amplifier's output can be derived by applying Kirchhoff's Current Law (KCL) and Ohm's Law to the circuit. Assuming an ideal op-amp, then no currents enter its inputs and the offset voltage between the non-inverting and inverting input should be equal to zero, meaning that the potential on both inputs are equal. KCL then gives

$$\frac{v_\mathrm{in}(t) - 0}{R_1} = C_\mathrm{F} \cdot \frac{\mathrm{d}v_C(t)}{\mathrm{dt}} = -C_F \cdot \frac{\mathrm{d}v_\mathrm{out}(t)}{\mathrm{dt}} \Leftrightarrow v_\mathrm{out}(t) = -\frac{1}{R_1 C_\mathrm{F}} \int v_\mathrm{in}(t)\ \mathrm{dt}. \tag{2.26}$$

The output voltage is evidently the integral of the input voltage, scaled by a negative factor $1/R_1C_F$. However, there is a drawback to this circuit configuration. The transfer function of this system is given by

$$H(\omega) = \left|\frac{V_{\text{out}}}{V_{\text{in}}}\right| = \frac{1}{\omega R_1 C_{\text{F}}}. \tag{2.27}$$

According to Eq. (2.27) when $\omega \to \infty$ then $H(\omega) \to 0$, and when $\omega \to 0$ then $H(\omega) \to \infty$. Hence, the integrator acts as an LP-filter with a constant -20 dB/decade roll-off towards the higher frequencies. At very low frequencies the capacitor acts as an open circuit, and the op-amp will approach its open-loop gain. While this is the frequency response sought after in the $\Delta\Sigma$ modulator, this is undesirable for any real-world application. It would result in the output being heavily saturated for lower frequency inputs, since the op-amp can only amplify the input up to the supply voltage (often a few volts lower than that).

Gain limitation can be introduced by connecting a feedback resistor R_{F} in parallel with the capacitor [9, p. 400], as shown in Fig. 2.9.

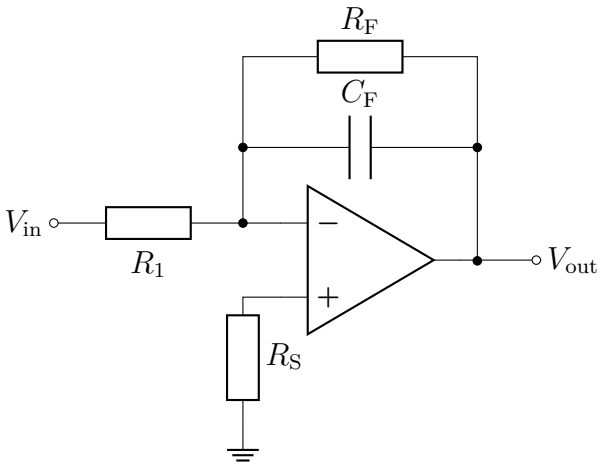

Figure 2.9: *Practical integrator circuit.*

Furthermore, a compensation resistor R_{S} is connected to the non-inverting input to ground in order to correct any potential voltage offset errors caused by the input bias currents, creating a so-called "lossy" or "practical" integrator [8, p. 643]. The rule of thumb is that this resistor value is given by R_{F} in parallel with R_1. Typically $R_{\text{F}} \gg R_1$, so $R_{\text{S}} \approx R_1$.

At lower frequencies the current will start flowing through R_{F}, effectively creating an inverting amplifier when the input is close to DC. This limits the low-frequency gain to

$$A_{\text{max}} = -\frac{R_{\text{F}}}{R_1} \tag{2.28}$$

and the new transfer function of the integrator circuit in the Laplace domain becomes

$$H(s) = -\frac{R_\mathrm{F}}{R_1} \cdot \frac{1}{1 + R_\mathrm{F} C_\mathrm{F} s}. \tag{2.29}$$

Eq. (2.29) shows that the integrator's frequency response no longer has a constant roll-off of -20 dB/decade, but will instead flatten out when below its cutoff frequency, determined by

$$f_\mathrm{low} = \frac{1}{2\pi R_\mathrm{F} C_\mathrm{F}}. \tag{2.30}$$

This frequency is important, since if the frequency is lower than f_low the circuit acts like an inverting amplifier, and no integration will occur. Hence, for the circuit to act as an integrator the input signal frequency needs to be greater than f_low. Therefore, f_low indicates the start of the integration range of the amplifier [9, p. 401].

At higher frequencies, the capacitor C_F is almost a short-circuit so the feedback resistor R_F can be disregarded. The transfer function is again given by Eq. (2.27), which gives the *unity-gain frequency*, the frequency for which the gain is equal to 1 (0 dB), as

$$f_\mathrm{high} = \frac{1}{2\pi R_1 C_\mathrm{F}}. \tag{2.31}$$

Any input frequency beyond f_high will be greatly attenuated, causing the output to approach zero. Hence, the new equation for the output becomes

$$v_{out}(t) = \begin{cases} -\frac{R_\mathrm{F}}{R_1} v_\mathrm{in}(t), & f < f_\mathrm{low} \\ -\frac{1}{R_1 C_\mathrm{F}} \int v_\mathrm{in}(t)\ \mathrm{dt}, & f_\mathrm{low} \leq f \leq f_\mathrm{high} \\ 0, & f \gg f_\mathrm{high}. \end{cases} \tag{2.32}$$

This, and an example comparison between a practical and ideal integrator's frequency response, can be seen in the bode plot in Fig. 2.10.

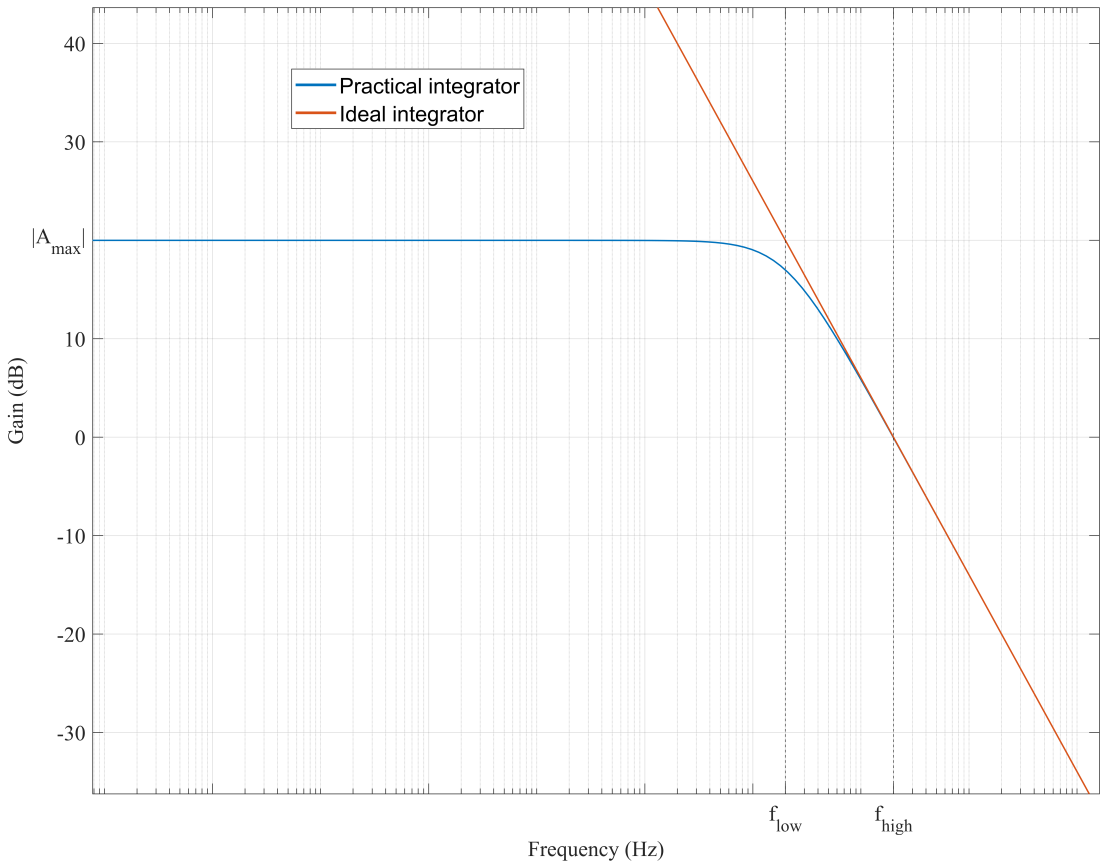

Figure 2.10: *Frequency response of the ideal and practical integrator.*

In conclusion, the useful integration range of the circuit is

$$f_{\text{low}} \leq f \leq f_{\text{high}} \Leftrightarrow \frac{1}{2\pi R_{\text{F}} C_{\text{F}}} \leq f \leq \frac{1}{2\pi R_1 C_{\text{F}}}. \tag{2.33}$$

The ratio R_1 and R_{F} sets the desired finite op-amp DC-gain, and in combination with C_{F} sets the useful integration range.

According to [2, p. 270], the finite gain can be an issue, but only for noise shaping below the corner frequency f_{low}. Therefore, if the signal band f_{B} is significantly larger than f_{low} the noise power is not significantly dependent on the DC-region of the frequency spectrum. This is due to the fact that frequencies close to f_{B} dominate the spectrum integral calculations. This desired cutoff frequency is related to the finite op-amp gain in the s-plane by

$$f_{\text{low}} = \frac{f_{\text{s}}}{2\pi} \ln\left(1 - \frac{1}{A_{\text{max}} + 2}\right) \simeq \frac{f_{\text{s}}}{2\pi(A_{\text{max}} + 2)}. \tag{2.34}$$

If $f_{\text{low}} \ll f_{\text{B}}$ then A_{max} does not affect the SNR substantially, so the OSR and gain must satisfy the condition

$$\pi(A_{\text{max}} + 2) \gg \text{OSR}. \tag{2.35}$$

Eq. (2.34) and Eq. (2.35) show that the op-amp's gain is dependent on the sampling frequency, and by correlation, the OSR. Selecting a reasonable value for f_{low} while satisfying the condition that it is much lower than the signal bandwidth limit ensures a suitable DC-gain that will not degrade the SNR significantly. However, the dynamic range of the integrator further limits the op-amp's performance. One needs to ensure that the selected gain will not result in an output voltage swing that exceeds the supply voltage of the op-amp, or else the resulting clipping of the output will add further error (noise) to the signal; degrading the SNR [2, p. 276].

The integrator in the $\Delta\Sigma$ ADC needs to be able take as an input the difference between the input signal (V_{signal}) and its quantized estimation from the feedback DAC (V_{DAC}). The circuitry above in Fig.2.10 would have trouble performing this task with its current design. It can, however, easily be expanded to include this mathematical operation. By splitting the non-inverting input into two inputs a summing integrator can be implemented (Fig. 2.11).

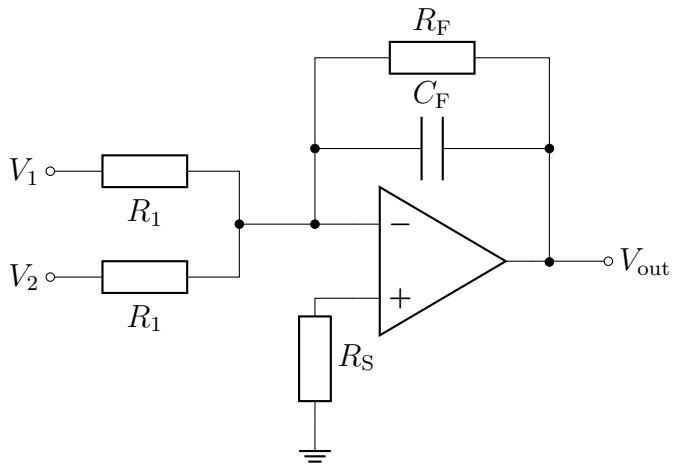

Figure 2.11: *Practical summing integrator circuit.*

Applying KCL and Ohm's Law once again and assuming an ideal amplifier, the sum of the currents going through the two resistors R_1 must be equal to the sum of currents going into R_{F} and C_{F}:

$$\begin{aligned} \frac{V_1}{R_1} + \frac{V_2}{R_1} &= -V_{\text{out}} \frac{R_{\text{F}} + \frac{1}{j\omega C_{\text{F}}}}{R_{\text{F}} \cdot \frac{1}{j\omega C_{\text{F}}}} = -V_{\text{out}} \frac{1 + j\omega R_{\text{F}} C_{\text{F}}}{R_{\text{F}}} \\ \Leftrightarrow H(s) &= \frac{V_{\text{out}}}{V_1 + V_2} = -\frac{R_{\text{F}}}{R_1} \cdot \frac{1}{1 + R_{\text{F}} C_{\text{F}} s}. \end{aligned} \tag{2.36}$$

Comparing Eq. (2.36) with (2.29) it becomes clear that the only distinction that can be made between the two is that V_{in} is replaced with $V_1 + V_2$. Furthermore, by assigning the voltages $V_1 = V_{\text{signal}}$ and $V_2 = -V_{\text{DAC}}$ and given Eq. (2.32) the output voltage can be described by

$$V_{out} = \begin{cases} -\frac{R_{\text{F}}}{R_1} \cdot (V_{\text{signal}} - V_{\text{DAC}}), & f < f_{\text{low}} \\ -\frac{1}{R_1 C_{\text{F}}} \int (V_{\text{signal}} - V_{\text{DAC}})\ \text{dt}, & f_{\text{low}} \leq f \leq f_{\text{high}} \\ 0, & f \gg f_{\text{high}}. \end{cases} \tag{2.37}$$

2.6.3 Comparator

Recalling from Sec. 2.3, the quantization block within the feedback loop of a 1-bit $\Delta\Sigma$ ADC can output a single bit of either a logic '1' or a logic '0', depending on the output from the integrator. This bitstream is fed into the FPGA, where it will be filtered and downsampled by the decimation filter. The pins of the *Arty S7* FPGA used will register a logic '1' if a voltage of +3.3 V is applied to them and a logic '0' if the voltage is zero [10]. This same bitstream is fed back into the system via the DAC and results in a two-level voltage; typically $\pm V_{\text{ref}}$. The advantage of such a 1-bit system is that both the ADC and DAC functionality can easily be implemented in the circuit using comparators or switches and provides the benefit of being inherently linear [2, p. 264]. An example circuit of a comparator can be seen in Fig. 2.12.

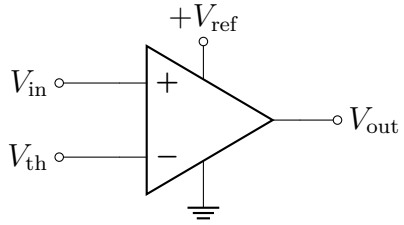

Figure 2.12: *Example comparator circuit.*

The comparator circuit in the figure above works by comparing an input V_{in} with a threshold voltage V_{th}. V_{out} is then decided by

$$V_{\text{out}} = \begin{cases} +V_{\text{ref}}, & V_{\text{in}} > V_{\text{th}} \\ 0, & \text{otherwise} \end{cases} \tag{2.38}$$

Note that not all comparators operate in the same fashion, and the description above is only meant to describe the functionality of the comparators used in this project. The precise configuration and specifications of the comparator used are discussed more thoroughly in Sec. 3.2.

All comparators have some amount of propagation delay; the output cannot instantaneously change when the input does. This is one of the main limitations the analog circuitry of the $\Delta\Sigma$ converter has on the selection of OSR. The comparator needs to have time to compare and output a new quantized value every sampling period. If the propagation delay is too long - or the OSR is too high - the benefits gained from the oversampling will be reduced. The result would be a worse SNR than the theoretical value.

2.6.4 Slew-rate

The finite slew-rate (SR) of the op-amps in the circuit is a severe limitation to overall conversion speed. By definition, the SR is the maximum slope of the output voltage

with respect to time. When the initial slope S_S of an input sinusoidal voltage is greater than the finite SR of the op-amp slew-rate, distortion occurs and the critical condition for non-distorted operation is therefore $S_S = \mathrm{SR}$, where the initial slope is defined as [11, p. 631]

$$S_S = 2\pi f V_p, \tag{2.39}$$

where f is the frequency of the signal and V_p is its amplitude. Hence, for the op-amps within a $\Delta\Sigma$ ADC to operate properly, the SR must be high enough to take as an input the high-frequency sampled signal without distorting the output. Then, by Eq. (2.39) and the critical condition for non-distorted operation, a correlation between minimum SR and sampling frequency f_s can be made:

$$SR_{\min} = 2\pi f_s V_p \tag{2.40}$$

2.7 Digital Decimation Filter

The $\Delta\Sigma$ ADC structure is completed with a moving average filter on the output of the modulator, realized using a digital decimation filter (or decimator) [3, p. 10]. In its most simple form, the decimator consists of a sharp digital LP-filter followed by a downsampler. The downsampler effectively "dumps" every nth sample of the signal to reduce the sampling rate and thus filtering out the high-frequency noise. In $\Delta\Sigma$ modulation the output can be downsampled by a factor of OSR, reducing the signal to the Nyquist-rate and, consequently, the number of samples that need to be processed digitally.

A common implementation of decimation filters in $\Delta\Sigma$ architectures is the *Cascaded Integrator-Comb* (CIC) filter. The popularity of CIC-filters in decimation applications is largely attributed to their computational efficiency [5]. Fig. 2.13 displays the most common way to realize a first order CIC-filter for downsampling.

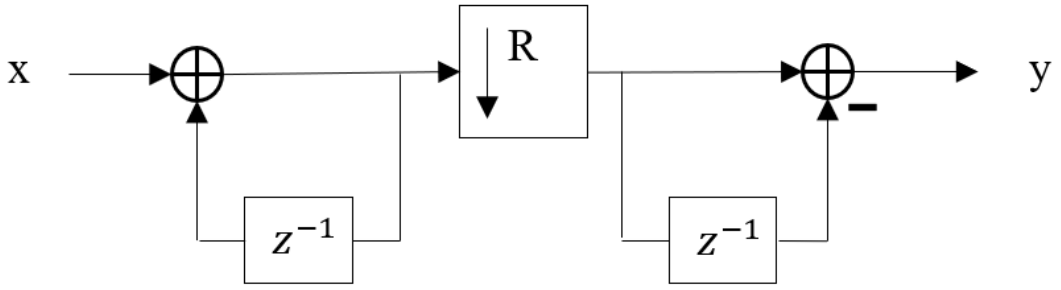

Figure 2.13: *First order CIC decimation filter.*

The structure in Fig. 2.13 is a cascade of an integrator-, a downsampling- and a differentiator (or comb)-stage. The main purpose of the CIC-filter is to provide anti-aliasing before the sample reduction. However, swapping the places of the

downsampling and comb sections reduces the amount of digital hardware operating at the high clock frequency, which in turn results in lower power consumption [5].

Increasing the order of the CIC-filter is done simply by cascading more integrators and comb sections. In $\Delta\Sigma$ applications, it is required for a modulator of order L to be followed by a filter of order $L+1$ in order to achieve sufficient anti-aliasing [2, p. 393].

For CIC-filters of order M, a passband gain of

$$G = (N \cdot R)^M, \tag{2.41}$$

is introduced [5]. Where N is the differential delay and R is the downsampling factor. This gain can be compensated for, in other words made equal to one, by binary right-shifting the output bit stream from the filter by

$$\log_2(G) = \log_2((NR)^M), \tag{2.42}$$

which applies when G is only allowed to be an integer of base two [5].

The high signal gain of the CIC-filter introduces a risk of arithmetic overflow [5]. However, this can be avoided by using subtraction with two's complement and ensuring that the registers have the bit-width of (at least) the maximum amplitude times the filter gain. This is accommodated for by using register widths of [5]

$$\text{Register width} = \text{Number of input bits} + M \cdot \log_2(NR). \tag{2.43}$$

2.8 I2S Serial Interface

I2S is a serial communication protocol initially developed by *Philips Semiconductors* [12]. The protocol needs 3 signals in order to transmit digital audio: serial bit clock (SCK), word select (WS) and serial data (SD). The SCK-signal controls the bitrate of the transmitted serial data while the WS-signal keeps track of the word lengths. I2S allows for stereo sound to be transmitted over the SD-line. If WS is low, then the data currently being transmitted over the SD-line is left-channel audio while WS being high indicates right-channel audio. The most significant bit of the data word is sent first.

The required bitrate for transmitting serial data can be calculated as

$$f_{\text{SCK}} = f_{\text{s}} \cdot r \cdot n, \tag{2.44}$$

where f_{s} is the sampling frequency of the transmitted signal, r is the bit resolution and n is the number of channels [13].

3

Design and Implementation

This chapter will explain the design process in detail. It is divided into three sections, with each section describing the design and simulation of each of the main parts for this project. Sec. 3.1 presents the block diagram in Simulink and relevant simulation data is displayed and discussed. In Sec. 3.2 a description of the implementation of the analog circuitry proposed in Sec. 2.6 is presented. Finally, the digital filter design process is described in Sec. 3.3.

3.1 Simulink Simulations

The discrete block diagrams for a first and second order $\Delta\Sigma$ modulator were drawn in Simulink and can be seen in Fig. 3.1 and Fig. 3.2. Note that the decimation filter is not yet included, so more in-band noise is expected.

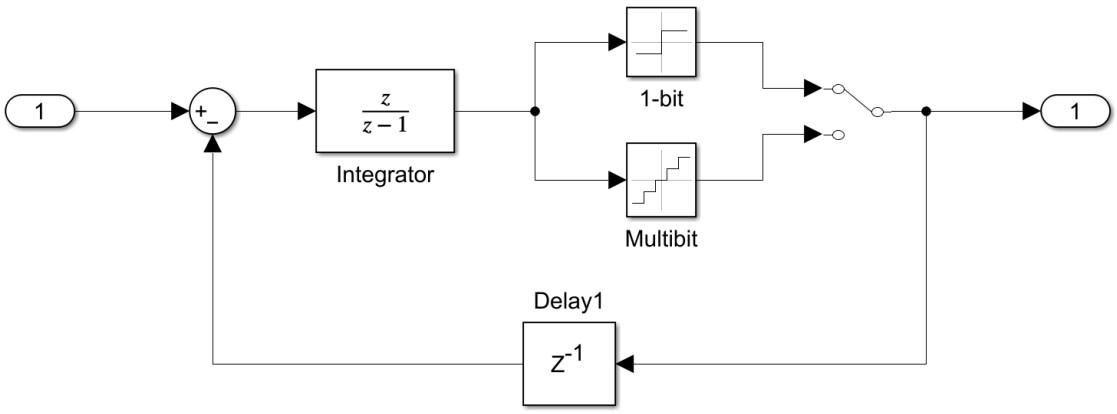

Figure 3.1: *First order modulator.*

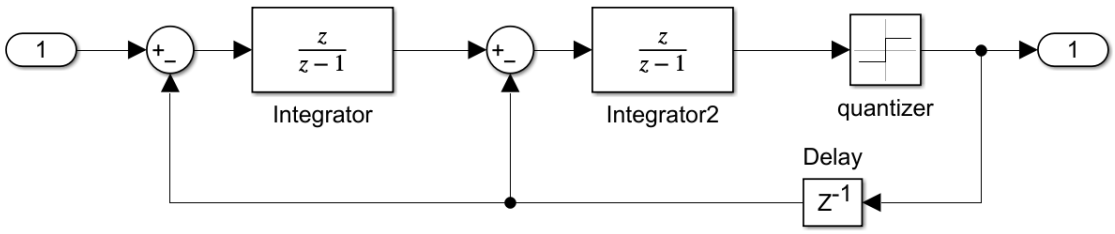

Figure 3.2: *Second order modulator.*

A sine wave with an amplitude of 0.85 V and 1.01 kHz was then used as an input to the modulators. The 1-bit quantization block had a threshold of 0 V with $V_{\text{ref}} = 1$ V. The reason for selecting the input amplitude as 0.85 V instead of V_{ref} was because of the fact that the peak SNR is achieved if the input amplitude is slightly lower than V_{ref}, around -6 dB [2, p. 64]. This is referred to as decibels relative to full-scale (dBFS). For -6 dB the input should be around 0.5 V, but simulations showed that 0.85 V was optimal for highest SNR.

Both block diagrams were simulated during 200 periods for some OSR of base 2. The quantizer in Simulink is essentially an ADC and DAC all in one and outputs $+V_{ref}$ if the integrator output is greater than 0, and $-V_{\text{ref}}$ it is lower. To fulfill the specifications from Sec. 1.4 the bandwidth was set to 20 kHz and to achieve anti-aliasing below 2 MHz, the OSR had to be high enough to ensure $f_{\text{s}} \geq 4$ MHz according to the Nyquist-Shannon sampling theorem. The power spectrum and SNR of pulse train output of the quantizer were calculated in a Matlab script and can be seen in Tab. 3.1. Using Eq. (2.21) and Eq. (2.22) the theoretical maximum values for the SQNR were also calculated and can be seen in Tab. 3.2.

Table 3.1: *SNR simulation result for first and second order* $\Delta\Sigma$.

1-bit ADC		
OSR	$\text{SNR}_{\mathbf{\Delta\Sigma,1}}$	$\text{SNR}_{\mathbf{\Delta\Sigma,2}}$
256	72.1 dB	100 dB
512	80.8 dB	114 dB
1024	89.8 dB	129 dB
2048	98.7 dB	145 dB

Table 3.2: *Theoretical SQNR for first and second order* $\Delta\Sigma$.

1-bit ADC		
OSR	$\text{SQNR}_{\mathbf{\Delta\Sigma,1}}$	$\text{SQNR}_{\mathbf{\Delta\Sigma,2}}$
256	68.9 dB	109 dB
512	77.9 dB	124 dB
1024	87.0 dB	139 dB
2048	96.0 dB	154 dB

A comparison of the above tables leads to the conclusion that the theoretical values differ from the simulation. This was expected because, as previously discussed, the linear noise model is a crude approximation to simplify calculations and should not be accepted at face value. The second order simulation shows around a 10 dB lower SNR in the simulated result. This displays clearly the effects of the quantization error harmonics discussed in Sec. 2.2. To verify this difference in SNR, the first order modulator's quantizer was removed and a white noise block was added to the signal path to create the linear model from Fig. 2.3. The linear model was then simulated with OSR = 256, which yielded an SQNR of 106 dB - very close to the approximation.

From Tab. 3.1 it is apparent that a second order modulator with an OSR of 256 achieves the minimum 100 dB SNR requirement. Furthermore, it achieved a THD of −82.1 dB which was also within the limitations.

The power spectrum of both a first order and second order 1-bit $\Delta\Sigma$ modulator can be seen in Fig. 3.3. The plot also verified the expected 20 dB/decade and 40 dB/decade noise shaping of the first and second order modulator respectively. Significant harmonics due to quantization could be observed on the second order plot at around frequencies 3 kHz, 5 kHz, and 7 kHz.

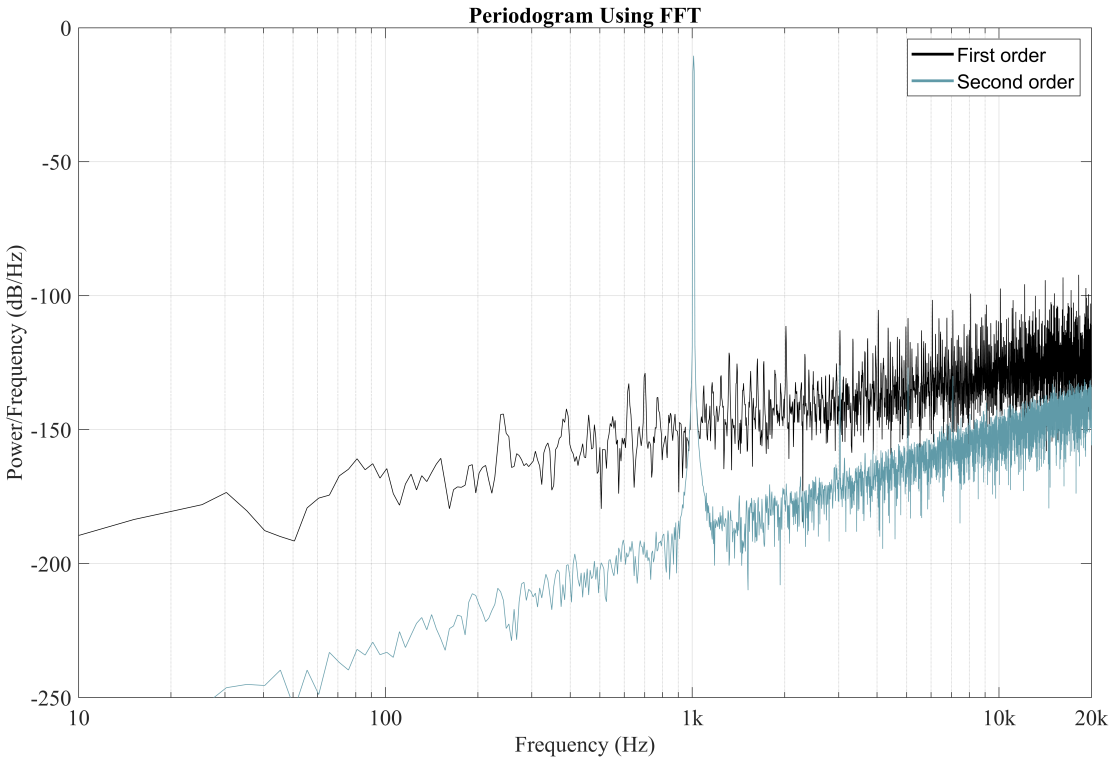

Figure 3.3: *Periodogram of first and second order 1-bit $\Delta\Sigma$ with OSR = 256.*

The integrator output can be seen in Fig. 3.4, and was used as a frame of reference when designing the integrator in the analog circuitry later on.

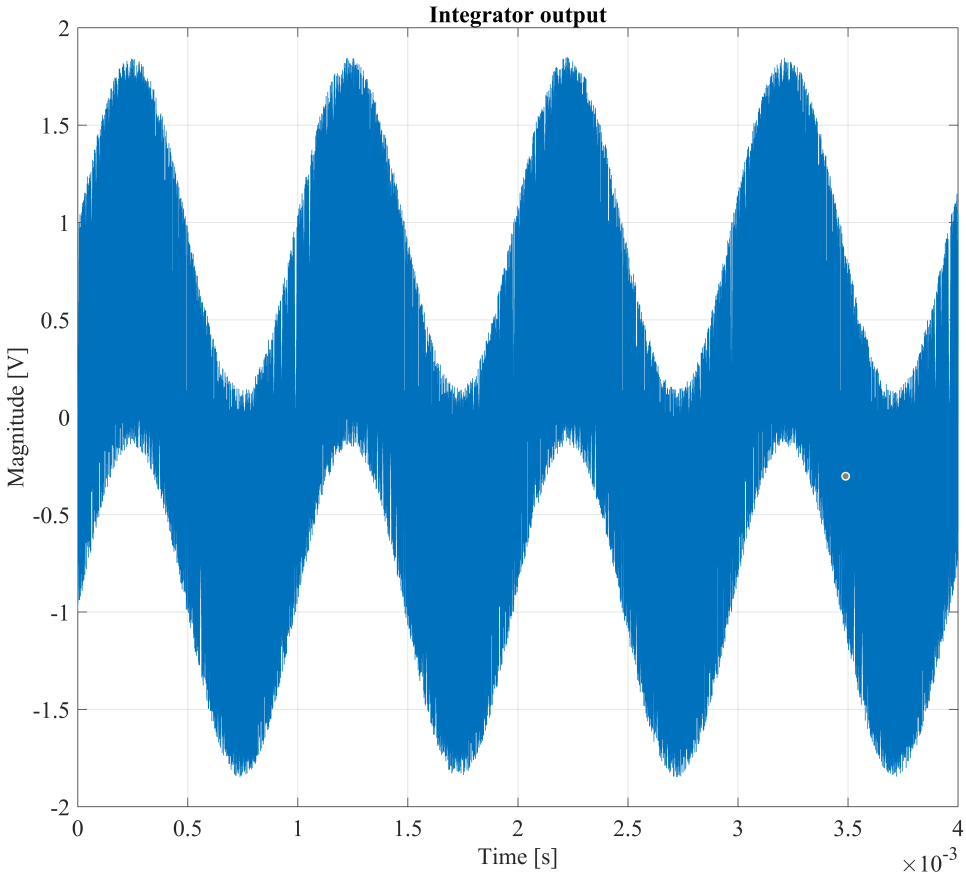

Figure 3.4: *Integrator output of the 1-bit* $\Delta\Sigma$ *with* $OSR = 256$.

A 3-bit first order modulator was also simulated in a similar fashion to observe the effects on the SNR, with the same OSR values. The output of the 3-bit quantizer can be seen in Fig. 3.5.

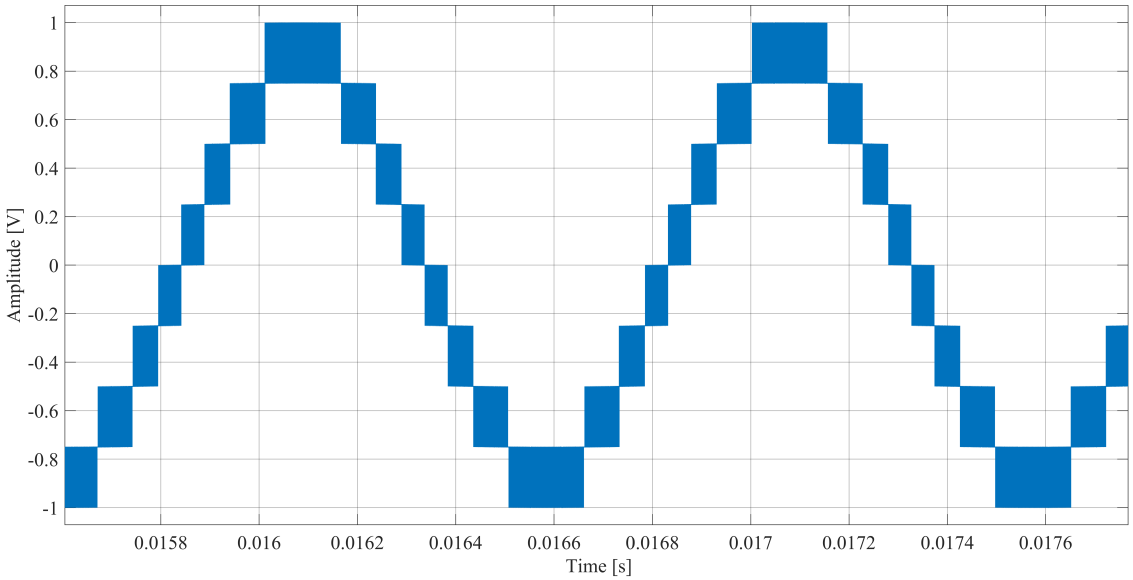

Figure 3.5: *3-bit quantizer output signal.*

With 3 bits there should be 8 DAC levels. However, it was discovered that the quantizer in Simulink operates to ensure that the sine wave is centered around 0 which, since 8 is an even number, requires an extra level; explaining the 9 levels shown in Fig. 3.5. Note that the number of quantization levels $k = 9$ implies $\log_2(9) \approx 3.17$ bits. The SNR results of the simulations can be seen in the second column of Tab. 3.3 as well as the theoretical SQNR values with $k = 9$ in the third column.

Table 3.3: *SNR simulation result for 3-bit first order* $\Delta\Sigma$.

3-bit ADC		
OSR	$\mathbf{SNR_{\Delta\Sigma,1}}$	$\mathbf{SQNR_{\Delta\Sigma,1}}$
256	86.3 dB	87.9 dB
512	94.6 dB	97.0 dB
1024	104 dB	106 dB
2048	113 dB	115 dB

Tab. 3.3 and Tab. 3.1 show an increase of about 14 dB SNR for each row using 3-bit instead of 1-bit quantization. The theoretical SQNR values are a bit higher, again showing that the linear model is only an approximation.

From Tab. 3.3, it is clear that a 3-bit first order modulator using OSR = 1024 could also achieve an SNR $\geq$ 100 dB. Additionally, it resulted in a THD of -97.9 dB. This gave the project two design options. Either a 1-bit second order modulator with an OSR = 256, or a 3-bit first order modulator with OSR = 1024. However,

the DAC non-linearities caused by the 3-bit quantization, as discussed in Sec. 2.2, would require advanced digital compensation circuits - which could be a whole thesis in itself. Furthermore, an OSR of 1024 leads to a sampling frequency of $f_s = 2 \cdot f_N \cdot \text{OSR} \approx 41$ MHz which is not a limitation for the *Arty S7* FPGA per se, because of its maximum clock frequency of 100 MHz [10]. However, it does increase the demand on the speed of the analog circuitry. It was therefore decided that a 1-bit second order modulator was preferable since the 1-bit quantization could easily be implemented with a single comparator while providing inherent linearity to the DAC. It also allows the DAC to simply be an output pin of the FPGA. In addition, the OSR could be as low as 256 which would relax the demand on the analog circuit.

3.2 Analog Circuit Design

The majority of the circuit design process was conducted in LTspice. To limit the number of unforeseeable factors that might negatively impact the circuit, the circuit was first implemented using as many ideal components as possible. These included the NMOS-transistor, resistors, capacitors as well as a separately constructed ideal op-amp consisting of a voltage-dependent voltage source (see Appendix A) which additionally provided the benefit of reducing the simulation time. These were replaced with the close to ideal "UniversalOpAmp2" included in the LTspice library, just to confirm that the constructed ideal OP amp worked as expected. Furthermore, all components were dimensioned with regards to an OSR of 256, giving $f_s \approx 10.24$ MHz. To follow a more modular approach, the S/H circuit was implemented first, followed by a first order architecture before finally implementing the second order converter.

3.2.1 S/H circuit

The S/H circuit presented in Sec. 2.6 was drawn up in LTspice, and can be seen in Fig. 3.6.

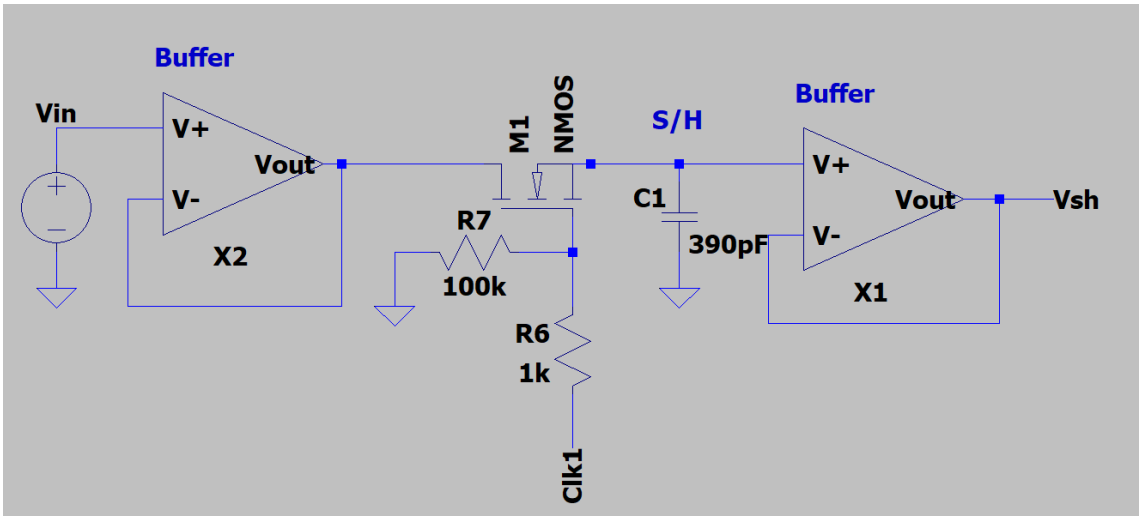

Figure 3.6: *Sample-and-hold circuit.*

The capacitor was sized by rewriting Eq. (2.24) with $\tau = T_s/2 \approx 48.83$ ns. A relatively high ESR of 20 Ω was assumed to ensure the minimum necessary capacitance. This resulted in a capacitance of approximately 390 pF. In Fig. 3.6 the previously mentioned ideal op-amp is connected as a buffer to the input and output of the circuit. A square wave generator of 0–5 V with a frequency equal to f_s was connected to the gate of the transistor to control the switching. A pull-down resistor (R_7) was added to the gate to ensure 0 V when the clock is low, as well as a resistor (R_6) connected in series to limit input current.

During simulation a sine wave with a peak value of 0.8 V and frequency of 10 kHz was used as input. The sampled voltage V_{sh} was plotted and observed. The sampling circuit proved functional for sampling up to the design frequency 10.24 MHz, indicating correct capacitor sizing. Fig. 3.7 demonstrates the sampling, where the sample V_{sh} is clearly held during the time when the clock signal is low. However, to more clearly show the step-like pattern of the sampled signal, a lower OSR of 128 is used in the figure. Also note, the clock signal amplitude is not accurate in the plot. The clock signal is simply scaled to fit so that the sampling can be observed in relation to the clock pulses.

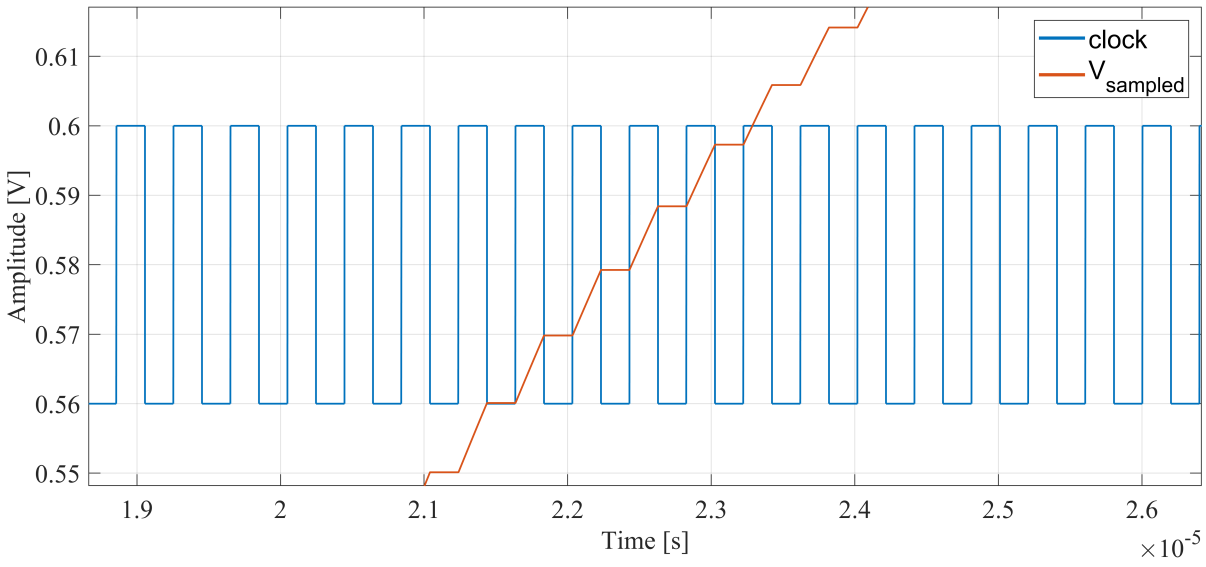

Figure 3.7: *S/H output with OSR* = 128.

3.2.2 First order circuit

The complete first order circuit design can be seen in Fig. 3.8.

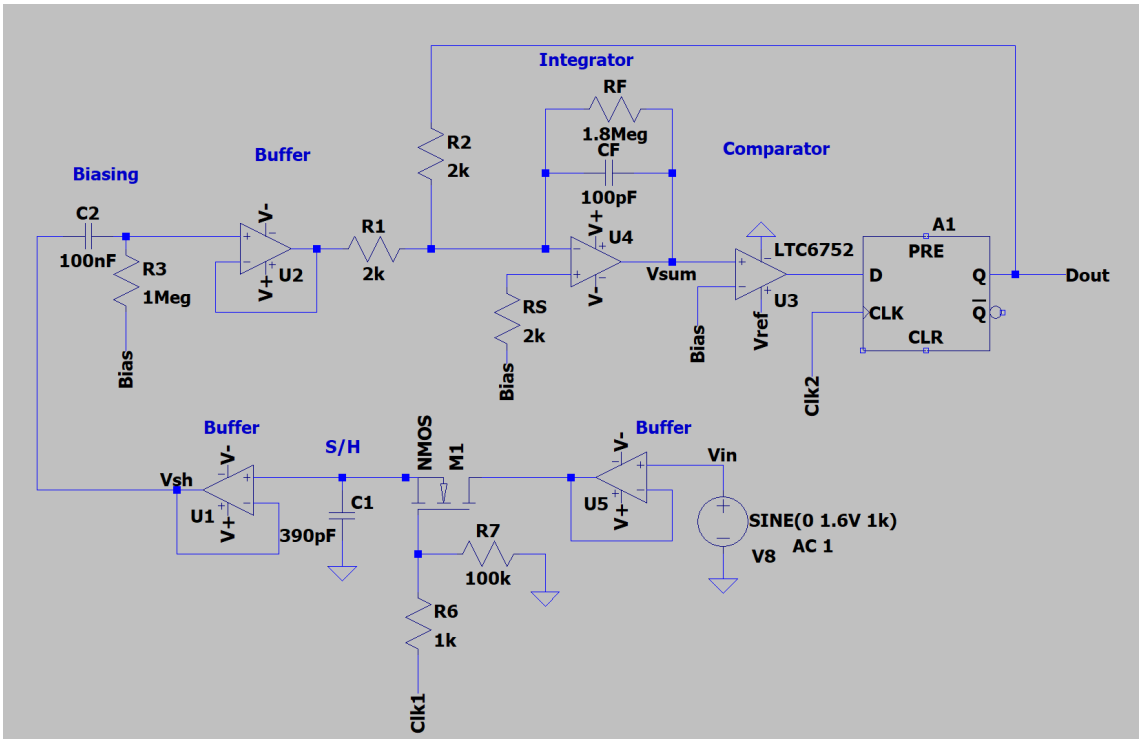

Figure 3.8: *First order circuit.*

The S/H circuit can be seen at the bottom of Fig. 3.8, which is then connected to the summing integrator circuit. The integrator's (U5) maximum DC-gain was calculated by rewriting Eq. (2.34) and choosing the upper corner frequency to be $f_{\text{low}} = 1$ kHz. This cutoff frequency was selected somewhat arbitrarily, but still making sure that the condition $f_{\text{low}} \ll f_{\text{B}}$ was fulfilled. This calculation gave a linear gain of $A_{\text{max}} \approx 1627$. Using the E12-series for resistors, selecting $R_{\text{F}} = 1.8$ MΩ and $R_1 = 1$ kΩ a gain of 1800 is yielded. R_1 needed to be low to push the upper limit frequency as high as possible. However, this resistance of R_1 later turned out to be too low which caused some issues for the output buffer, like high currents and an increase of the THD, and was set to 2 kΩ instead (see Sec. 3.2.4). This gave a gain of 900, or 59 dB. R_{S} and R_2 were, as previously argued for, set equal to R_1.

The capacitor value was calculated to 88 pF by rewriting Eq. (2.30) and using the selected R_1 and $f_{\text{low}} = 1$ kHz. The capacitance was rounded up to the E6-series value of 100 pF, resulting in the integration range 884 Hz – 795.8 kHz. Ideally, the integration range would go up to f_{s} but this would have required either a very small R_1, resulting in too high currents, or such a small capacitor that it would be close to the same range of parasitic capacitances in the circuit - which would not be feasible.

The comparator is an LTC6752 from *Analog Devices* [14] which was selected due to its fast propagation speed of 2.9 ns and high toggle rate. Its supply voltage was

$V_{CC} = V_{ref} = 3.3$ V and $V_{EE} = 0$ V to accommodate the FPGA voltage on its pins. The output of the comparator was connected to a D flip-flop (DFF) to simulate the FPGA input and output pins. This DFF was clocked on the same clock as the sampling circuit, but phase-shifted 180° to ensure that each quantized sample only is outputted when the S/H circuit is "holding" a sample. It was set to be able to output 0 V or 3.3 V just like the FPGA pins and connected in feedback to the summing integrator.

The input of the integrator in the block diagram in Fig. 3.1 is $V_{in} - 1$ if in the previous iteration the integrator output was greater than 0 V, and $V_{in} + 1$ if it was lower. Since a summation connection is used in the analog circuit, the subtraction would have to be implemented by addition. Meaning adding a negative voltage when the quantizer outputs a logic '1' and a positive voltage for a logic '0'. This would require some switching circuit or another comparator, with additional supply voltages of ±1 V. Furthermore, the input voltage to the LTC6752 must be between $V_{EE} - V_{CC} = 0 - 3.3$ V [14], meaning no negative voltage can be applied. Comparison with 0 V with the integrator output centered around 0 V was therefore not possible with this comparator. Instead, a virtual ground replaced most of the actual ground of the modulator circuit, adding a DC-bias of $V_{ref}/2 = 3.3/2$ V $= 1.65$ V to the circuit (see Fig. 3.8). A first order passive HP-filter was also added to the input to separate the AC- and DC-signals. A high resistance of 1 MΩ was selected to ensure a low DC-current, and with a 100 nF capacitor everything below 1.5 Hz is blocked.

The threshold voltage of the comparator was also set to 1.65 V. This ensured that the comparator could output 0 V when its input was lower than 1.65 V and 3.3 V if it was higher. Recall that this integrator design actually outputs the inverted integration of the input, therefore the comparator is actually an inverted version of the quantizer in Fig. 3.1. Hence, when the comparator's output is 0 V it is equivalent to the ideal integrator's output in Fig. 3.1 being above 0 V so a negative voltage should be added via the feedback. However, since a DC-bias of 1.65 V has been added to this circuit, and the fact that the comparator is comparing the integrator output to this voltage, adding 0 V has the same effect. Similarly 3.3 V is added when the integrator's output is above 1.65 V (ideal integrator's output is below 0 V). Thus, the correct functionality of integration of differences is achieved. It is however important to make sure that the capacitor C_2 is fully charged to the bias voltage before connecting it to the rest of the circuit, or else the integrator will race towards its maximum DC-gain.

3.2.3 Second order circuit

The first order circuit in Fig. 3.8 was expanded by adding an inverting amplifier with a linear gain of $A = 0.5$, and a second summing integrator connected in feedback to create a second order modulator, which can be seen in Fig. 3.9.

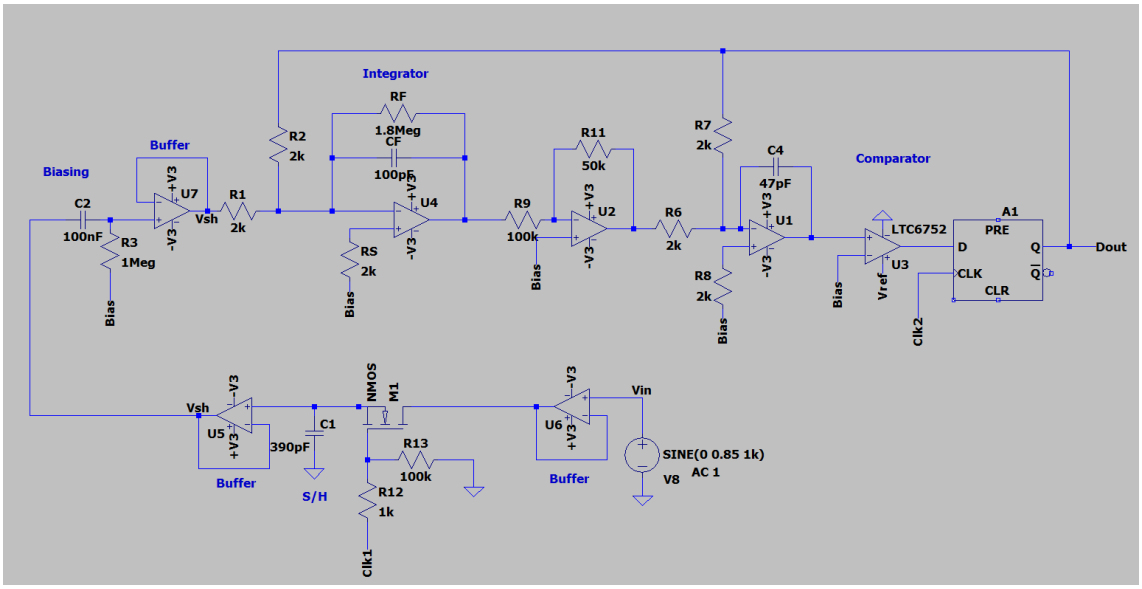

Figure 3.9: *Second order circuit.*

Since the first integrator inverts the signal, the inverting amplifier was added after it to invert the output back again and thus providing the same operation as the block diagram in Fig. 3.2, as well as lowering its voltage swing which enables higher gain in the second integrator. The second summing integrator was connected with open-loop gain and half the capacitance of the first integrator, to further push the integration interval to frequencies closer to f_s in order to more efficiently integrate the error.

3.2.4 Hardware implementation

To fully implement the design on a perf board or PCB, real components have to be carefully selected.

For the integrator and buffers *Analog Devices* AD829 video op-amp was deemed suitable [16], due to its wide bandwidth, high SR, and high gain-bandwidth product (GBP). In the datasheet it was specified that the lowest THD was achieved when a load resistance of 2 kΩ was used, hence the resistor values selected for the summing circuit of the integrators. Multi-layer ceramic capacitors of 390 pF and 100 pF were selected for the sampling circuit and integrator respectively. An N-channel MOSFET from *Onsemi* was selected as a switch [17], due to its reasonable threshold voltage and low on-resistance, yielding low switching loss. The resistors can be Surface Mount Device (SMD) resistors to save space.

It is worth mentioning that the sine wave in the LTspice simulations has no noise in it. A real-world signal generator would generate a signal containing some noise. This, combined with the fact that the buffers and other components add extra thermal noise as well, makes it unlikely that the simple sampling system would be able to accurately hold a sample each sampling period, given the high sampling frequency. It might require rigorous testing and design changes. The noise would be so high in

relation to the sampling steps that each sample would be hard to distinguish. This explains why most S/H ICs available off-the-shelf have a relatively low maximum operating frequency, and why SC-integrators are common practice for achieving the sampling and integration in DT $\Delta\Sigma$ ADCs.

3.3 Digital Hardware Design

The digital hardware was implemented on the *Spartan 7*: XC7S25-CSGA324 FPGA using VHDL to describe the logic. The top-level design can be found in Appendix B. Behavioral testing was done in Modelsim while synthesis and implementation were carried out using Vivado.

3.3.1 Clock generation

The *Arty S7* FPGA board has a 12 MHz and a 100 MHz crystal oscillator directly available on pins. However, the 100 MHz oscillator can be used to drive phase-locked loops (PLLs) in order to generate clocks of various frequencies. This function can be accessed through Vivado by using the "Clocking Wizard" IP core. This creates a component that can be instantiated in a VHDL program, taking the 100 MHz as an input and generating a clock with a chosen frequency as an output. The Clocking Wizard resource was used to generate the 10.24 MHz sampling clock.

The PLLs can only be used to generate frequencies down to 4 MHz. So, in order to generate the downsampled 40 kHz clock, a synchronous counter was created. The counter increments at every rising edge of the sample clock up to OSR = 256 and forces a trig-signal to go high. At the next sample-clock edge, the trig-signal is forced low while the counter is reset. This process repeats continuously. The generated trig-signal was used as the Nyquist-rate clock.

Since the filter output has a sampling frequency of 40 kHz and 25 bits resolution, then by Eq. (2.44) a 1 MHz clock is required to serially transmit the data over one channel using the I2S protocol. However, since the system clock is not an even multiple of 1 MHz, 32 bits were used instead yielding a frequency of 1.28 MHz. This was again achieved using a synchronous counter.

Tab. 3.4 shows a summary of the different clocks used in the design.

Table 3.4: *Clock table.*

Clock name	Frequency [MHz]	Decimation factor
clk_hf	10.24	-
clk_sck	1.28	8
clk_lf	0.04	256

3.3.2 Decimation filter

The filter was implemented using simple digital building blocks such as adders and registers. By the reasoning in Sec. 2.7, and since a second order modulator was used, a third order filter was designed based on the block diagram in Fig. 3.10.

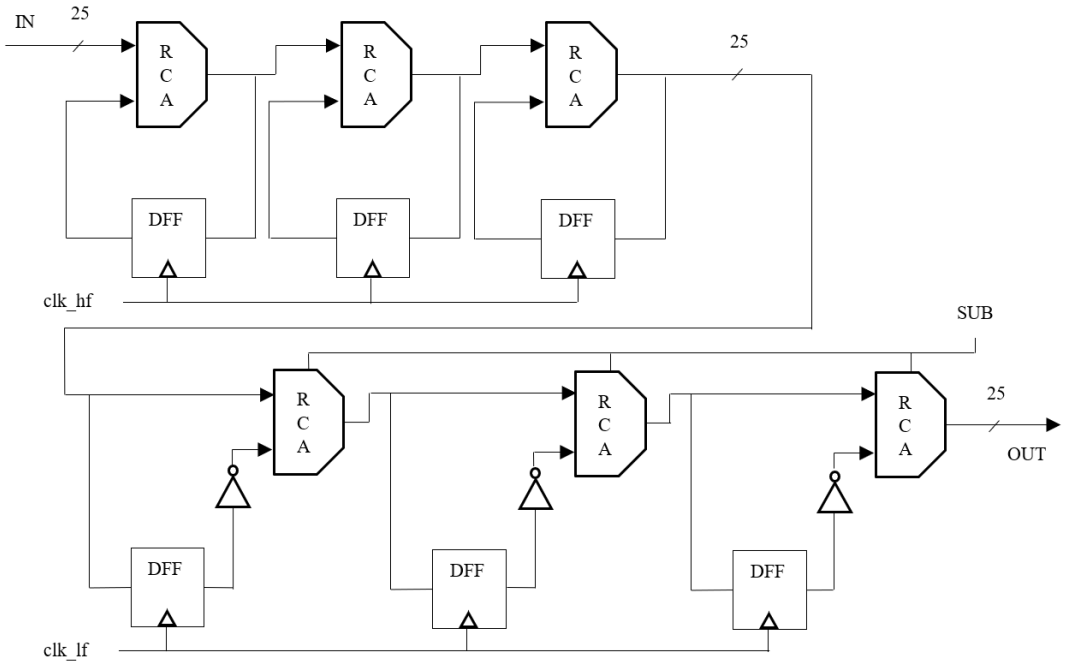

Figure 3.10: *Hardware realization of decimation filter.*

The input (IN) in Fig. 3.10 is 1 bit, however, by Eq. (2.43) the bit-width of the registers needs to be 25 bits. Therefore, the input must be sign-extended to 25 bits. The addition and subtraction operations were implemented using ripple-carry adders (RCAs). For the adders to perform subtraction using two's complement, the subtraction signal (SUB) is set to '1' and sent as a carry-in to the RCAs while the second operand is negated.

The memory elements (25-bit registers) in the circuit are composed of positive edge-triggered DFFs. Separate clocks were used to trigger the DFFs in the integrator and comb sections, where clk_hf is the high-frequency sample clock and clk_lf is the downsampled Nyquist-rate clock as described in the previous section.

Compensation for the filter pass-band gain, described in Eq. (2.42), was done by interpreting the output bits as a fixed-point number. That is, instead of arithmetically right shifting the output, which would result in loss of information, it was converted to fixed-point s1.23 format. This means that bit 25 is interpreted as a sign bit, bit 24 as an integer bit followed by 23 bits of fraction. Comparing this to the input shows a 23-bit increase in resolution.

A modular design was employed when implementing the system in VHDL. The different blocks of Fig. 3.10 were combined into modules that were instantiated as components in the top-level design (see Appendix B).

3.3.3 Serial communication

The I2S serial bus was implemented as a shift register, shifting one bit of an input data word to the left at every rising of the serial clock, clk_sck, and writing the MSB to the output. The hardware implementation of an n-bit shift register is shown in Fig. 3.11.

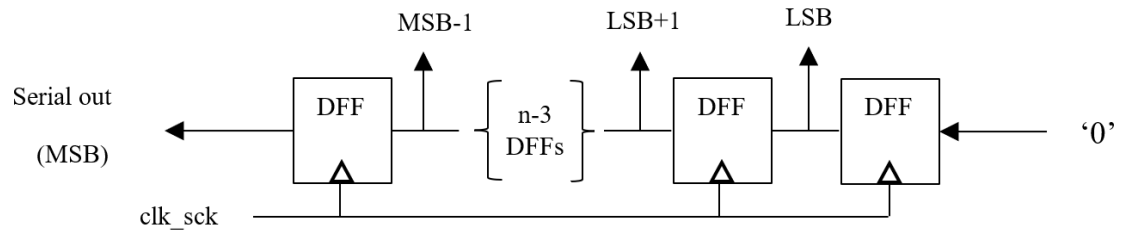

Figure 3.11: *Hardware implementation of an n-bit shift register.*

At each clk_lf pulse, a new data word is loaded to the bit lines. Then, as Fig. 3.11 shows, a left shift is performed at every clk_sck pulse. The register is padded with zeros from the least significant bit (LSB). Furthermore, the WS-signal is initialized to zero at reset and inverted at every clk_lf pulse.

4
Results

4.1 Circuit Simulations

The first order circuit from Fig. 3.8 was simulated using an input sine wave with different amplitudes, ranging from 0.85 – 1.6 V, and 1 kHz frequency with $f_s = 10.24$ MHz during 100 periods. Since $V_{ref} = 3.3$ V and not 1 V in this circuit - following the reasoning in Sec. 3.1 - an input voltage of 1.6 V should be selected for -6 dBFS. However, changing the amplitude did not significantly change the SNR, although a lower amplitude improved the THD.

An ideal inverting amplifier with 0 dB gain (to correct the 180° phase shift) in series with a first order passive LP-filter with a cutoff frequency of 20 kHz was added to the output of the modulator to demodulate the PCM-output of the DFF to see if it was actually retrieving the sampled signal. In the figures below, the sampled voltage (V_{sh}) with an amplitude of 1.6 V together with the filtered output (V_{filter}) is seen in Fig. 4.1 and the integrator output is seen in Fig. 4.2. In the figures below, the sampled voltage (V_{sh}) with an amplitude of 1.6 V and the filtered output (V_{filter}) can be seen in Fig. 4.1 and the integrator output is shown in Fig. 4.2.

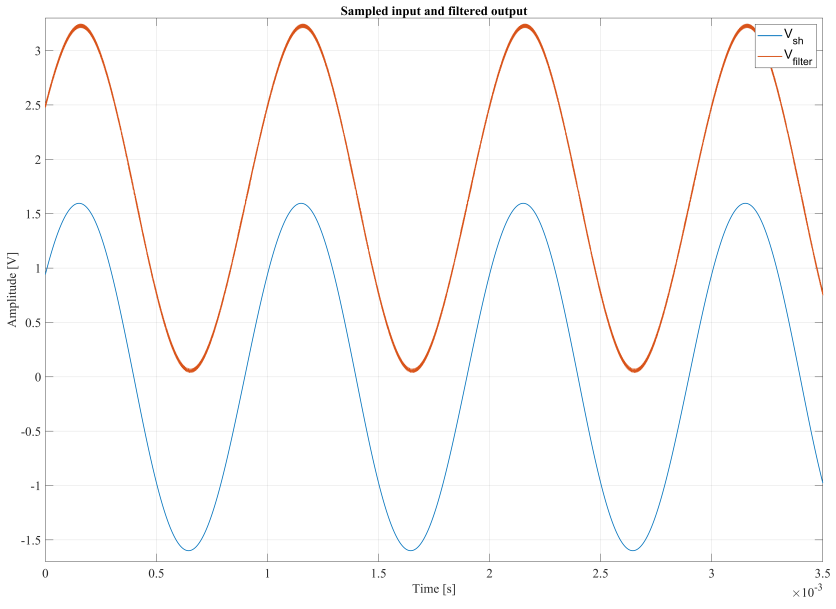

Figure 4.1: *Simulated results of the sampled voltage and the filtered output.*

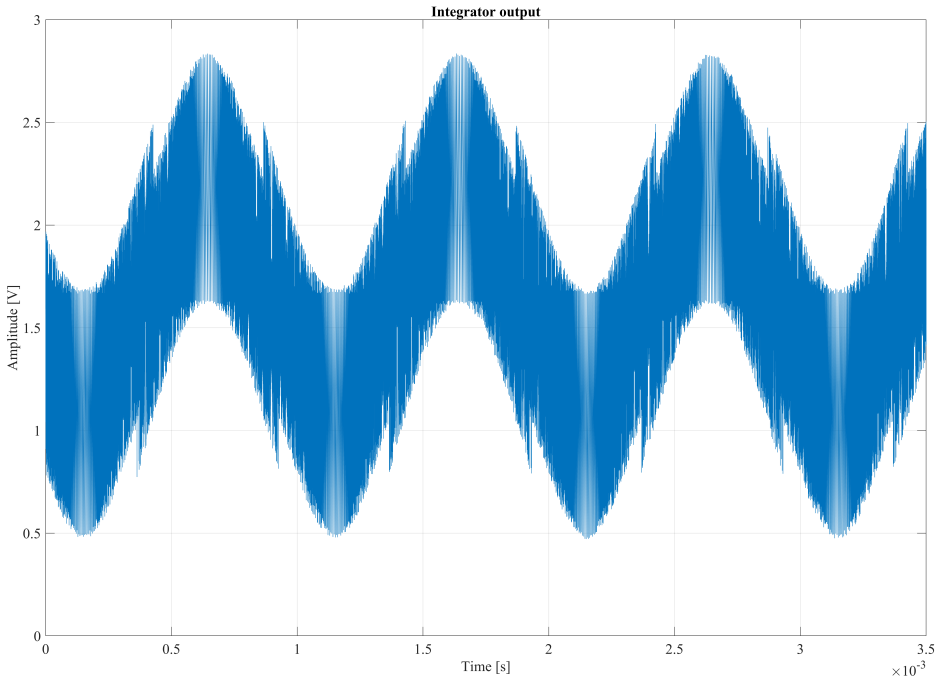

Figure 4.2: *Integrator output.*

The filtered voltage clearly shows - although more noisy, with a slight voltage drop - the recreated sampled signal with a 1.65 V DC-bias. It also had a phase shift of 38°. The integrator output had an appearance similar to that of the Simulink simulation in Fig. 3.4, and a dynamic range of approximately 2.3 V; meaning no chance of clipping occurring, as well as the voltage being low enough for the input of the comparator.

The FFT of the LP-filtered signal was plotted with a Hamming windowing function in LTspice and can be seen in Fig. 4.3.

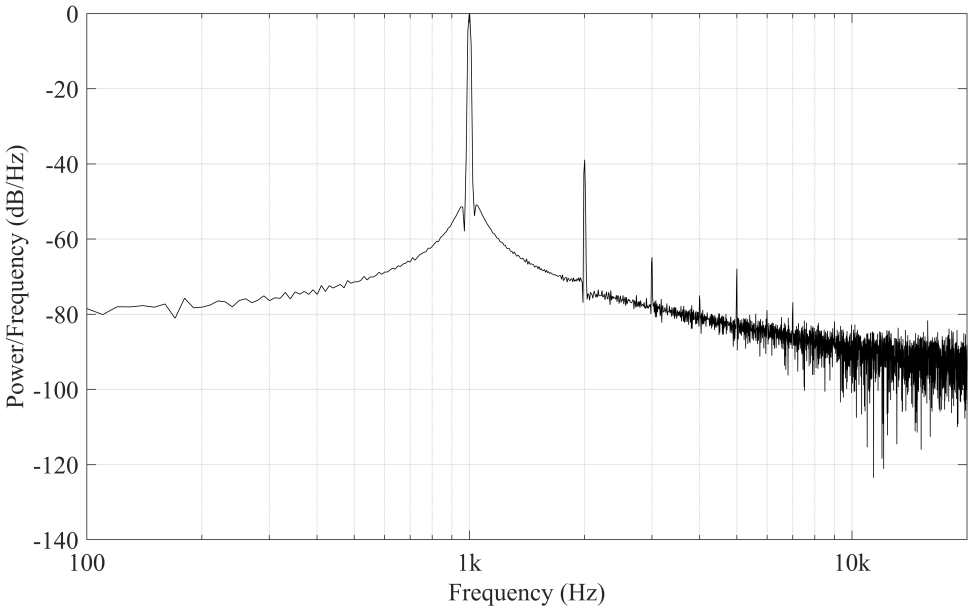

Figure 4.3: *Power spectrum of first order circuit with OSR* = 256 *after 100 periods.*

The SNR was calculated using the FFT-data in Matlab. The plot above shows harmonics at 2, 3, 4, 5 and 7 kHz; the 2 kHz being of the most significant amplitude. The simulation yielded an SNR of 38.3 dB, which was significantly lower than hoped for. However, the spectrum below the fundamental 1 kHz frequency component shows a 20 dB/decade noise shaping slope, starting from -80 dB. Lowering the input amplitude to 0.8 V could increase the SNR to 40 dB. The vast difference between this SNR and the one from the Simulink simulation is largely due to the prominent harmonics. The harmonics were surprisingly large and led to a THD of -18.8 dB. It is suspected that this might have been caused by some error in the algorithms LTspice utilizes, although this is not confirmed. The spectrum of the output from the DFF showed that the majority of the noise was pushed outside the signal band. However, it did not show the expected 20 dB/decade noise shaping above the signal band.

The transfer function of the ideal integrator in the Simulink model was replaced with the discretized transfer function of the practical integrator (Eq. (2.36)), with the resistance and capacitance values calculated in Sec. 3.2.2. This was then simulated with an OSR of 256 during 100 periods to more accurately compare the SNR result from Simulink and LTspice, which yielded an SNR of 74.7 dB; showing that the limited DC-gain did not significantly impact performance.

The second order circuit from Fig. 3.9 was also simulated using the same simulation parameters as with the first order circuit. Unfortunately, this circuit yielded very similar results as the first order circuit; achieving an SNR of only 43 dB, with the same harmonics affecting the noise performance. The FFT did not show the expected 40 dB/decade noise shaping. Thus, it was concluded that this circuit design was not functional as a second order $\Delta\Sigma$ modulator.

The circuit was altered in various ways, including moving the summing circuitry and the inverting amplifier into a differential amplifier instead to see if it might change the performance. The gain and capacitance values of the integrators were also changed in different ways to see if the issue could be resolved there. However, no changes tested could provide a solution to this circuit, rendering it not suitable for real-world implementation.

It was instead postulated that the functioning first order circuit could be used for the final design, but with a much higher OSR of 2048. Since the simulation results in Tab. 3.1 of Sec. 3.1 showed SNR = 98.7 dB, perhaps combined with the digital decimation filter, a better SNR could be possible to achieve in the laboratory.

4.2 CIC-filter

In order to test the filter, the block-diagram in Fig. 3.10 was first implemented in Simulink and connected to the output of the second order modulator in Fig. 3.2. Depicted in Fig. 4.4 is the Simulink model of the filter.

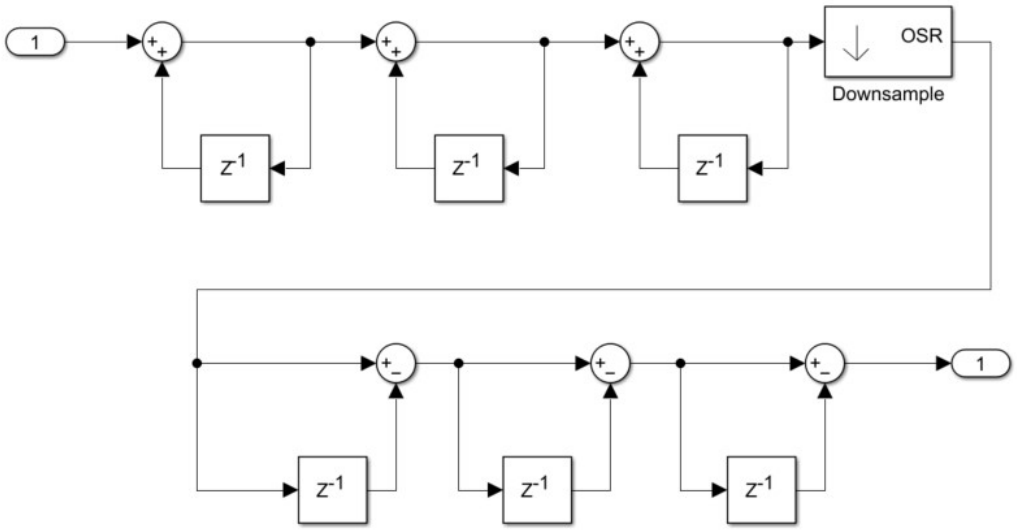

Figure 4.4: *Simulink model of CIC decimation filter.*

Simulations using a 1.01 kHz sine wave with an amplitude of 0.7 (-3 dBFS) resulted in the power spectrum shown in Fig. 4.5.

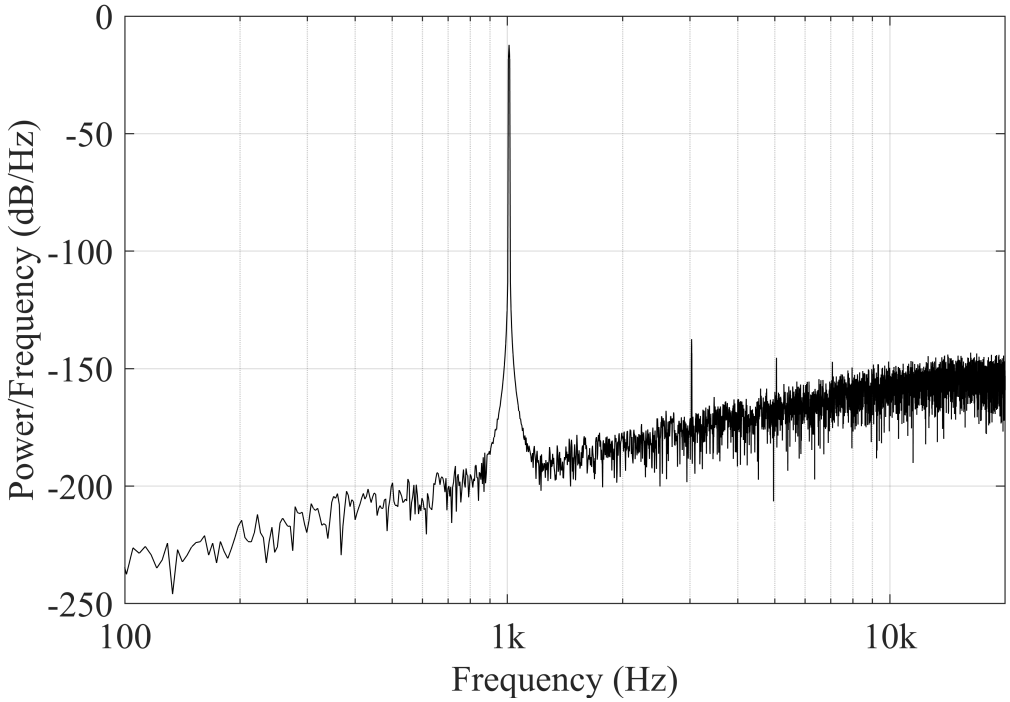

Figure 4.5: *Power spectrum of digital filter output.*

Calculating SNR and THD in Matlab from the spectrum in Fig. 4.5 yielded about 107.5 dB and -120 dB respectively, which are promising results.

After confirming the functionality and performance in Simulink, the filter design was implemented in VHDL and tested in Modelsim. A testbench was created in VHDL to provide stimuli to the input ports. Behavioral testing of the filter was carried out by generating a look-up table (LUT) of 10240 sine wave samples and converting them to a PCM-signal to be used as input to the design. A behavioral model of a first order modulator was created in order to generate the input signal. Downsampling the input signal of 10240 samples by a factor of OSR = 256 yields an output sine wave with 40 samples per period, which was confirmed through simulation. The sine wave generated from the LUT and output from the filter are shown in Fig. 4.6.

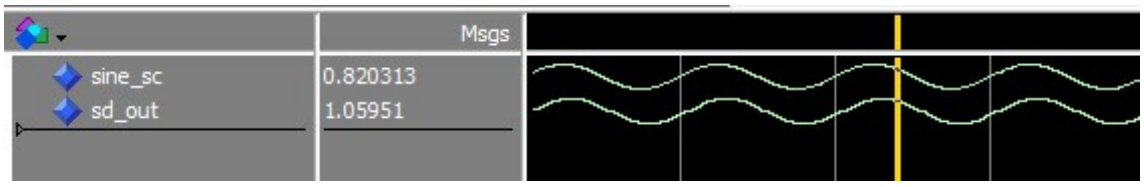

Figure 4.6: *Sine wave from LUT amd digital filter output.*

As shown in Fig. 4.6, the delay of the filter introduces a slight phase-shift between the two signals. The insufficient resolution of Modelsim's wave-window hides the fact that the output contains significantly fewer samples than the sine wave from the LUT. However, thorough examination of the output reveals that it is properly

downsampled, which can also be confirmed by using an LUT with fewer samples. After the behavioral testing in Modelsim was completed, the design was successfully synthesized and implemented on the target device in Vivado. Analyzing the timing reports showed a worst-case delay of 78.192 ns. This is acceptable since it is less than the period of the fastest clock in the system (clk_hf), which has a period of 97.65625 ns.

Finally, the serial interface was successfully implemented in VHDL using the I2S protocol. This enables easy communication with an audio analyzer for future hardware testing.

5

Conclusions and Recommendations

The Simulink simulations showed that in order to achieve the required specifications of SNR $\geq$ 100 dB and THD ≤ -80 dB within the 20 kHz bandwidth, a 1-bit second order $\Delta\Sigma$ modulator with a minimum OSR of 256 was optimal. It was also possible to reach 104 dB SNR by instead using a 3-bit first order modulator with 1024 OSR, although it would require more work to mitigate the imposed non-linearity of the multi bit DAC.

The proposed circuit designed to implement the modulator proved only to be functional for a first order modulator, although performing significantly worse than the simulated ideal model. It could recreate the modulated signal using a LP-filter to demodulate it, but only achieving an SNR of 38.3 – 40 dB and THD of -18.8 dB; the harmonics being the main degrading factor.

The circuit was simulated using ideal components, and it was deemed unlikely that the S/H circuit would prove functional using real components due to thermal noise making the fast samples indistinguishable from the noisy signal. The sampling circuit simulated did also not include parasitic capacitances, which would no doubt be an issue in a real-world implementation. However, more testing in a laboratory environment is needed before disregarding the design completely. SC-integrators would most likely be a better choice to achieve the sampling, and could be implemented using discrete components although requiring a more complex circuit. Another option is to instead investigate the possibility of implementing a CT $\Delta\Sigma$ modulator, which was not covered in this report.

The digital CIC-filter performed well in simulation and was able to synthesize on the target device. However, improvements can be made to the design. The addition and subtraction operations were performed using RCAs. This was unnecessary since the target device was an FPGA. It would be more efficient to let the FPGA resolve this logic and optimize these operations. The RCAs add a lot of delay to the signal path, which increases proportionally to the number of bits. In the case of this filter, the operands are 25 bits which contributes to a significant portion of the total delay. Furthermore, the register does not need to be 25 bits wide throughout the whole design since the signal is successively amplified. Optimizing the register-widths would save area and reduce power consumption.

Since the analog hardware implementation of the modulator did not function properly, the filter was not tested using an actual FPGA. This could be included in future work. Tests using an off-the-shelf $\Delta\Sigma$ modulator, another FPGA with a synthesized digital modulator or, ideally, a finished prototype of the proposed design as input to the filter could be performed.

The chosen architecture of a 1-bit quantization, second order modulator is still believed to be the best option in terms of performance in relation to circuit complexity. Implementing this modulator using SC-circuits in combination with a third order CIC-filter would result in a far cheaper design than off-the-shelf ADCs with similar precision and speed.

Bibliography

[1] D. Hafizovic, M. Karlsson, "GaN Transistor Based Digital Class-D Amplifier with Global Feedback", MSc thesis, Department of Electrical Engineering, Chalmers University of Technology, Gothenburg, Sweden, 2020. [Online]. Available: https://odr.chalmers.se/items/3c9bd263-cca0-4d69-ba70-219974ded528, Accessed on: 2023-05-06

[2] F. Maloberti, *Data Converters*, Dordrecht, The Netherlands: Springer, 2007.

[3] S. Pavan, R. Schreier, *Understanding Delta-Sigma Converters*, 2nd ed., Hoboken, NJ, USA: John Wiley & Sons, Inc., 2017. [Online]. Available: https://ieeexplore.ieee.org/book/5264508, Accessed on: 2023-05-06.

[4] L. Bengtsson, *Elektriska Mätsystem och Mätmetoder*, Lund, Sweden: Studentlitteratur AB.

[5] R. Lyons, "A Beginner's Guide To Cascaded Integrator-Comb (CIC) Filters", *DSP related*. [Online]. Mar. 26, 2020. Avaliable: https://www.dsprelated.com/showarticle/1337.php (accessed on: 2023-05-06).

[6] B. Karlström, *Kretsanalys*, 2nd ed., Lund, Sweden: Studentlitteratur AB, 2017.

[7] L. Bengtsson, B. Karlström, *Transformer och Filter*, 2nd ed., Lund, Sweden: Studentlitteratur AB, 2016.

[8] P. Scherzm S. Monk, *Practical Electronics for Inventors*, 4th ed., USA: McGraw-Hill Education, 2016.

[9] J.M. Fiore, *Operational Amplifiers & Linear Integrated Circuits: Theory and Application*, 3rd ed., Utica, NY, USA: J.M. Fiore, 2021. [Online]. Available: https://www2.mvcc.edu/users/faculty/jfiore/OpAmps/OperationalAmplifiersAndLinearICs_3E.pdf, Accessed on: 2023-05-06.

[10] *Reference Manual*, Pullman, WA, USA: Digilent, 2019. [Online]. Available: https://digilent.com/reference/programmable-logic/arty-s7/reference-manual, Accessed on: 2023-05-06.

[11] A.P. Malvino, *Electronic Principles*, 5th ed., USA: McGraw-Hill, 1993.

[12] "I²S," *Wikipedia*. [Online]. Mar. 25, 2023. Available: https://en.wikipedia.org/wiki/IÂšS#cite_note-i2s-spec-1. (accessed on: 2023-06-05).

[13] *Inter-IC Sound Bus (I2S)*, San Jose, CA, USA: Cypress Semiconductor Corporation, 2016. [Online]. Available: https://www.infineon.com/dgdl/Infineon-Component_I2S_V2.0-Software%20Module%20Datasheets-v02_07-EN.pdf?fileId=8ac78c8c7d0d8da4017d0ea2ed3c2596

[14] *LTC6752/LTC6752-1/LTC6752-2/LTC6752-3/LTC6752-4 - 280 MHZ, 2.9ns Comparator Family with Rail-to-Rail inputs and CMOS Outputs*, USA: Linear Technology Corporation, 2014. [Online]. Available: https://www.analog.com/media/en/technical-documentation/data-sheets/6752fc.pdf, Accessed on: 2023-05-14.

[15] M. West, "Ideal Op Amp Model", *YouTube*, Jan. 8, 2018. [Video]. Available: https://www.youtube.com/watch?v=t8vR499dEg8&ab_channel=MaryWest

[16] *High Speed, Low Noise Video Op Amp*, Norwood, MA, USA: Analog Devices, Inc., 2011. [Online]. Available: https://www.analog.com/media/en/technical-documentation/data-sheets/AD829.pdf, Accessed on: 2023-05-22.

[17] *Field Effect Transistor - N-Channel, Enhancement Mode*, Semiconductor Components Industries, 2010. [Online]. Available: https://www.mouser.se/datasheet/2/308/1/MMBF170_D-2315979.pdf, Accessed on: 2023-05-22.

A

Ideal Op-Amp Schematic

The design in Fig. A.1 was implemented based on [15].

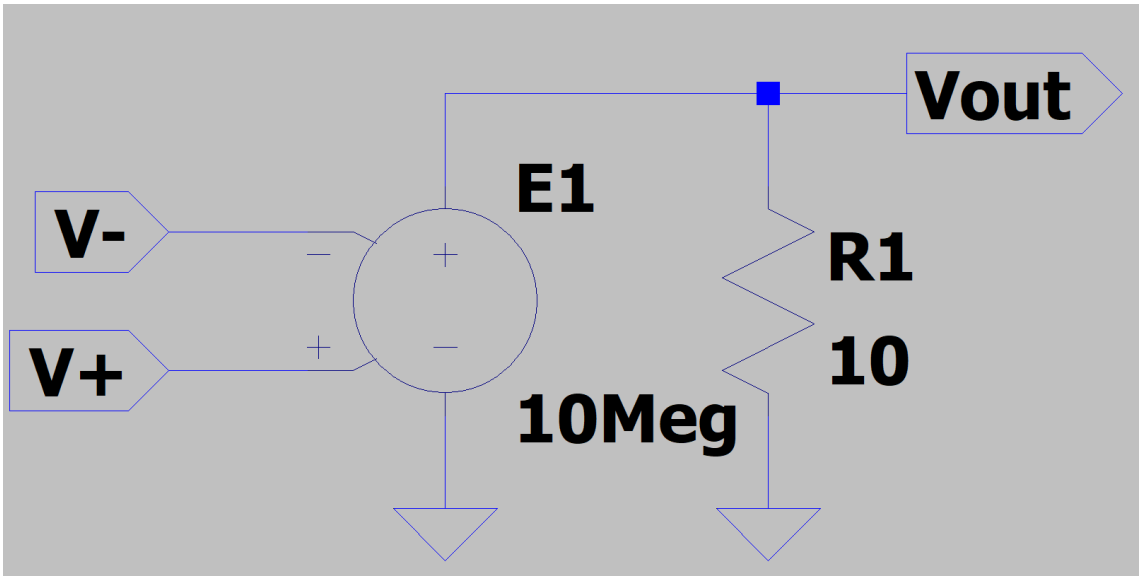

Figure A.1: *Ideal op-amp modeled as a voltage dependant voltage source.*

B

Top Level VHDL-code

```
1  LIBRARY ieee;
2  USE ieee.std_logic_1164.ALL;
3  USE ieee.numeric_std.ALL;
4
5  ENTITY decfilter IS
6    GENERIC (
7      reg_width : INTEGER := 25;   -- bit-width of registers
8      order     : INTEGER := 3;    -- CIC-filter order
9      OSR       : INTEGER := 256); -- oversampling ratio
10   PORT (
11     clk_sys : IN  STD_LOGIC;  -- 100 MHz system clock
12     reset   : IN  STD_LOGIC;  -- hardware reset low
13     adc_in  : IN  STD_LOGIC;  -- input from adc
14     sd_fb   : OUT STD_LOGIC;  -- feedback signal
15     bit_clk : OUT STD_LOGIC;  -- serial bit clock (I2S)
16     fs_clk  : OUT STD_LOGIC;  -- frame sync (I2S)
17     sd_out  : OUT STD_LOGIC;  -- serial data (I2S)
18     clk_hfo : OUT STD_LOGIC); -- sample clock to SnH-circuit
19 END decfilter;
20
21 ARCHITECTURE structural OF decfilter IS
22
23   TYPE arr IS ARRAY (0 TO 2 * order) OF signed(reg_width - 1
       DOWNTO 0);
24
25   SIGNAL count   : INTEGER;
26   SIGNAL count2  : INTEGER;
27   SIGNAL clk_hf  : STD_LOGIC; -- 10.24 MHz sample clock from
       PLL
28   SIGNAL clk_lf  : STD_LOGIC; -- 40 kHz downsampled clock
29   SIGNAL clk_sck : STD_LOGIC; -- 1.28 MHz serial clock
30   SIGNAL x_arr   : arr;
31
32   COMPONENT integrator
33     GENERIC (d_width : INTEGER);
34     PORT (
35       clk    : IN  STD_LOGIC;
36       reset  : IN  STD_LOGIC;
37       in_hf  : IN  signed(d_width - 1 DOWNTO 0);
```

```
38        out_hf : OUT signed(d_width - 1 DOWNTO 0));
39    END COMPONENT;
40
41    COMPONENT differentiator
42      GENERIC (d_width : INTEGER);
43      PORT (
44        clk    : IN  STD_LOGIC;
45        clk_lf : IN  STD_LOGIC;
46        reset  : IN  STD_LOGIC;
47        in_lf  : IN  signed(d_width - 1 DOWNTO 0);
48        out_lf : OUT signed(d_width - 1 DOWNTO 0));
49    END COMPONENT;
50
51    COMPONENT clk_wiz_0
52      PORT (
53        resetn   : IN  STD_LOGIC;
54        clk_in1  : IN  STD_LOGIC;
55        clk_out1 : OUT STD_LOGIC);
56    END COMPONENT;
57
58    COMPONENT shift_reg
59      GENERIC (b_width : INTEGER);
60      PORT (
61        clk     : IN  STD_LOGIC;
62        clk_lf  : IN  STD_LOGIC;
63        bit_clk : IN  STD_LOGIC;
64        reset   : IN  STD_LOGIC;
65        data_i  : IN  signed(b_width - 1 DOWNTO 0);
66        shift_o : OUT STD_LOGIC;
67        fs_clk  : OUT STD_LOGIC);
68    END COMPONENT;
69
70  BEGIN
71
72    -- generate sample clock --
73    clk_gen : clk_wiz_0
74    PORT MAP(reset, clk_sys, clk_hf);
75
76    -- generate feedback --
77    PROCESS (clk_hf, reset)
78    BEGIN
79      IF reset = '0' THEN
80        sd_fb <= '0';
81      ELSIF rising_edge(clk_hf) THEN
82        sd_fb <= adc_in;
83      END IF;
84    END PROCESS;
85
86    -- generate low frequency clock --
```

```
87  PROCESS (clk_hf, reset)
88  BEGIN
89    IF reset = '0' THEN
90      clk_lf <= '0';
91      count  <= 0;
92    ELSIF rising_edge(clk_hf) THEN
93      IF count = OSR - 1 THEN
94        clk_lf <= '1';
95        count  <= 0;
96      ELSE
97        clk_lf <= '0';
98        count  <= count + 1;
99      END IF;
100   END IF;
101 END PROCESS;
102
103 -- generate serial bit-clock --
104 PROCESS (clk_hf, reset)
105 BEGIN
106   IF reset = '0' THEN
107     clk_sck <= '0';
108     count2  <= 0;
109   ELSIF rising_edge(clk_hf) THEN
110     IF count2 = 7 THEN
111       clk_sck <= '1';
112       count2  <= 0;
113     ELSE
114       clk_sck <= '0';
115       count2  <= count2 + 1;
116     END IF;
117   END IF;
118 END PROCESS;
119
120 -- sign-extend --
121 x_arr(0) <= (0 => '1', OTHERS => '0') WHEN (adc_in = '1')
      ELSE (OTHERS => '1');
122
123 -- generate integrator side (high frequency) --
124 G1 : FOR i IN 0 TO order - 1 GENERATE
125   int : integrator
126   GENERIC MAP(reg_width)
127   PORT MAP(clk_hf, reset, x_arr(i), x_arr(i + 1));
128 END GENERATE;
129
130 -- generate comb side (low frequency) --
131 G2 : FOR i IN 0 TO order - 1 GENERATE
132   dif : differentiator
133   GENERIC MAP(reg_width)
```

```
134     PORT MAP(clk_hf, clk_lf, reset, x_arr(order + i), x_arr(
            order + i + 1));
135   END GENERATE;
136
137   -- generate serial data and frame sync for I2S interface --
138   I2S : shift_reg
139   GENERIC MAP(reg_width)
140   PORT MAP(clk_hf, clk_lf, clk_sck, reset, x_arr(2 * order),
          sd_out, fs_clk);
141
142   -- clock signals to output --
143   bit_clk <= clk_sck;
144   clk_hfo <= clk_hf;
145
146 END structural;
```

Department of Electrical Engineering
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden
www.chalmers.se

CHALMERS