Master's Thesis

MA684

Efficient VHDL Implementation of Decimation Filters for a Next Generation Wireless Receiver

Ralph Gruber

Institute for Electronics
Graz University of Technology
Head: Univ.-Prof. Dipl.-Ing. Dr. techn. Wolfgang Pribyl

Supervisor: Univ.-Ass. Dipl.-Ing. Dr. techn. Peter Söser
External Supervisor: Dipl.-Ing. Andreas Lessiak

Graz, May 2011

Kurzfassung

Durch die rasche Entwicklung von Kommunikationstechnologie in den vergangenen Jahren hat auch der Bedarf an Systemen mit digitaler Signalverarbeitung zugenommen. In immer mehr Bereichen kommen Mixed-Signal-Schaltungen, die sowohl analoge als auch digitale Signale verarbeiten, zur Anwendung. Für hochauflösende $\Sigma\Delta$ A/D-Umsetzer werden Dezimierungsfilter benötigt, welche die Abtastfrequenz verringern und Rauschen ausfiltern. Dieser an sich einfache Vorgang nimmt in einer integrierten Schaltung, bei der Größe und Stromverbrauch eine dominierende Rolle spielen, an Komplexität zu.
Ziel dieser Arbeit ist es, unterschiedliche Dezimierungsfilter für den Einsatz in einem linearen Empfänger zu untersuchen. Ein $\Sigma\Delta$ A/D-Umsetzer mit 14 Bit Auflösung liefert dabei das digitale Datensignal. Darauf basierend werden verschiedene Filtertypen mit besonderen Eigenschaften diskutiert. Zwei effiziente Hardware Implementierungen werden vorgestellt sowie sämtliche Schritte vom Entwurf bis zur Verifikation beschrieben. Um die korrekte Funktionalität der Filter zu verifizieren und deren Verhalten zu vergleichen wurde eine Testumgebung aufgesetzt. Dort werden mit Hilfe von SNR Analysen die Implementierungen evaluiert. Um auch eine Größenabschätzung der Filter zu erhalten, wurden diese für eine 130 nm CMOS Technologie synthetisiert.

Abstract

Digital signal processing in communication technology has strongly evolved in the recent years. The need for highly efficient and power aware digital systems has increased as their field of application has enlarged. Mixed signal systems with A/D and D/A converters are implemented in many devices nowadays. For high precision $\Sigma\Delta$ A/D converters, decimation filters are required for noise filtering and sampling rate reduction. What sounds simple in the first instance, is an challenging task for integrated systems where size and power consumption matters.

Therefore the objective of this work is to evaluate decimation filters for application in a linear receiver architecture. A $\Sigma\Delta$ A/D converter with 14 bits resolution delivers the digital input data to the decimation filter. Different types of decimation filters with various properties are compared and two efficient VHDL implementations are presented. Thereby design considerations as well as hardware implementation issues are discussed in detail. Finally a test bench for verification and evaluation of the implemented filters is presented. In order to compare the implementations regarding to their efficiency, SNR simulations are performed. To get an estimation of the required chip area the filters are synthesized for a 130 nm CMOS technology.

STATUTORY DECLARATION

I declare that I have authored this thesis independently, that I have not used other than the declared sources / resources, and that I have explicitly marked all material which has been quoted either literally or by content from the used sources.

....

date (signature)

Acknowledgement

This thesis was written from October 2010 to May 2011 at Institute for Electronics at Graz University of Technology in cooperation with Infineon Technologies Austria.

First I would like to thank my parents for supporting me during my years of study and making it possible for me to follow my own interests.
I want to thank my external supervisor Andreas Lessiak who supported me during my work at Infineon and always had good advice for my thesis.
Furthermore I have to thank my supervisor Peter Söser for his fast and uncomplicated help, especially at the end of my thesis.
Last but not least I want to thank all my friends and colleagues for making the years of study in Graz an unforgettable time with great memories.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

This work is developed in the context of an European Union project called *Pollux* where research for automotive electronics is expedited. *Infineon Technologies* is taking part in the project with a concept for a new linear wireless receiver for automotive applications. In a modern car there are several dozens of electronic control units communicating with each other. Rain sensors, window openers and temperature sensors are only some of the components mounted in a vehicle. In order to reduce costs and weight, wireless communication is preferred more and more over conventional wire lines.

A linear receiver architecture offers several advantages compared to a non-linear architecture. First of all lower bandwidths can be used by means of digital filtering which leads to higher sensitivity as well as higher ranges. Because of it's linear architecture this type of receiver is suitable for other modulation schemes like QAM or multi-level FSK. This makes the receiver suitable for a wider field of applications. Another advantage due to the digital implementation of filters is a lower production spread, no need for calibration and less test time.

The conversion from analog to digital signals requires high resolution in order to maintain the signal information. High sampling rates and large bit-widths make the digital signal processing a challenging task. A fixed point hardware implementation of digital filters involves the risk that overflows occur or that coefficient quantization leads to improper behavior. By carefully choosing proper filter architectures and bit-widths these problems can be circumvented. At the same time chip size as well as power consumption should be kept low. In a multi-rate digital signal processing chain, early decimation is the key to a power efficient implementation.

1.2 Objectives

Linear receiver architectures require the implementation of high resolution A/D converters in order to maintain the full signal information in the digital domain. In this project a $\Sigma\Delta$ A/D converter with 14 bits resolution is used. ADC properties like noise shaping and high oversampling require the application of a decimation filter. Reduction of sampling rate and noise are the most important features of this filter.

The main objective of this work is to find an efficient hardware design for a decimation filter under consideration of chip area and power consumption. There are specifications for the receiver which have to be satisfied by the implementation. Because the decimation filter is the first digital signal processing block in the receiver it's performance is influencing the whole digital processing chain.
For the filter implementation VHDL should be used and the correct functionality has to be verified by means of simulations. For this reason a test bench was set up which allows various simulations of the VHDL code.

Figure 1.1, which shows the frequency spectrum of a ΣΔ A/D converted signal, illustrates the process of decimation filtering. Due to the noise shaping property of the ADC, quantization noise is moved to higher frequencies. Therefore the signal within the band of interest can easily be separated from noise by low-pass filtering. The high oversampling of the signal is indicated by f_s which is much higher than the Nyquist frequency of the band of interest f_{pass}. Therefore it is reduced to a lower sampling frequency f_{s_new}. The resulting signal still contains the whole information but got rid of unnecessary noisy frequency components.

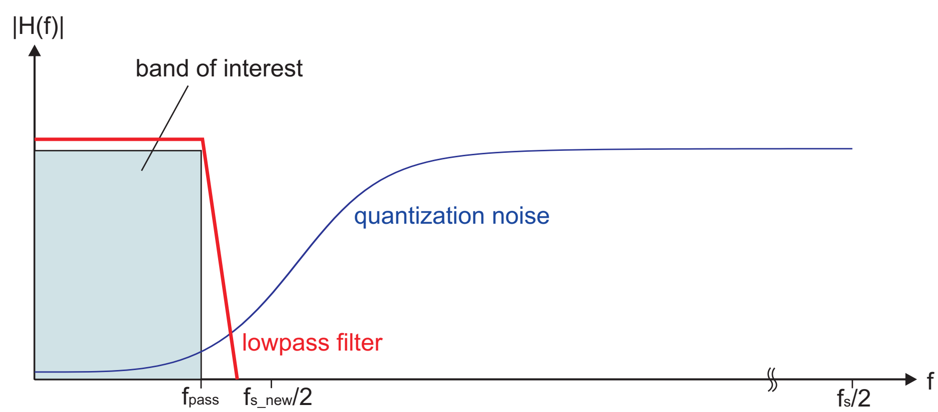


Figure 1.1: Low-pass filtering of a ΣΔ A/D converted signal

1.3 Outline

The main work starts with chapter 2 where literature research is done. In this chapter ΣΔ A/D converters and their properties are discussed. This is important to get an understanding for the need of decimation filters. At the end of the chapter an overview of power consumption in the CMOS technology is given.

In chapter 3 the architecture and the specifications of the linear receiver are presented. They build the basis for the filter concept. In section 3.4 various filter types and design

methods are discussed. The focus is concentrated on filter structures particularly suitable for rate decimation as well as efficient hardware implementations.
The verification process of the implemented filters is summarized in chapter 4. Test bench, setup and method are described. Additional functions used for simulation purposes are explained as well.

Chapter 5 describes the design process and implementation of two different decimation filters. They were implemented in VHDL and simulated with the test bench introduced before.
In chapter 6 the simulation results of both implementations are compared with each other in order to get to a conclusion which structure is more efficient. Area and power estimations as well as SNR simulations are performed.
At the end of the thesis in chapter 7 a summary and outlook is done.

Chapter 2

Literature research

2.1 ΣΔ A/D Converters

Analog-to-digital conversion is a key feature in nowadays communication systems and signal processing applications. Signals are converted from the analog domain to the digital for several reasons.

- Digital data is easier to process and store
- Digital signals are insensitive to environmental influences
- Digital signal processing is more flexible than analog processing
- Digital signals are not influenced by component tolerances or temperature coefficients
- Mostly digital filters need less chip area than their analog counterparts

In this section the advantages of ΣΔ A/D converters, particularly continuous time implementations, are investigated. Below there is a comparison of A/D converter architectures in terms of resolution and signal bandwidth.

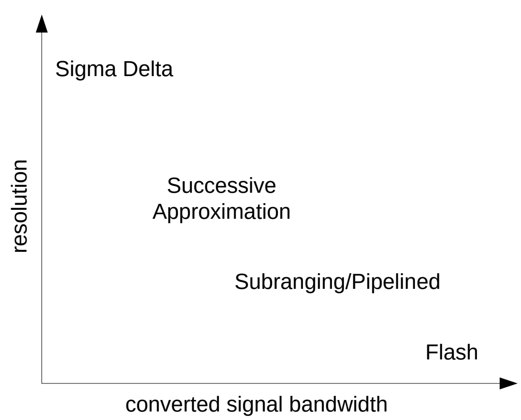


Figure 2.1: Resolution vs. signal bandwidth of different A/D converter architectures [4]

From figure 2.1 it can be seen that $\Sigma\Delta$ converters are best suited for low sampling rates and high resolution conversion whereas flash converters have low resolution at high sampling rates. So $\Sigma\Delta$ ADCs are mainly deployed for high precision signal conversion of e.g. measured sensor data. Also in communication systems they are widely spread because of their high performance and low power implementations. The need for high performance A/D converters comes with a higher level of integration and higher degree of digitalization. By moving digital signal processing of receivers more and more towards the antenna, the performance requirements of the ADC grow. The demands on linearity and dynamic range are increasing. At the same time power consumption should be kept low to fulfill the requirements of battery based mobile devices.

In figure 2.2 the structure of a sigma delta converter is depicted, it is composed of the following components:

- Integrator
- Quantizer
- 1 bit DAC

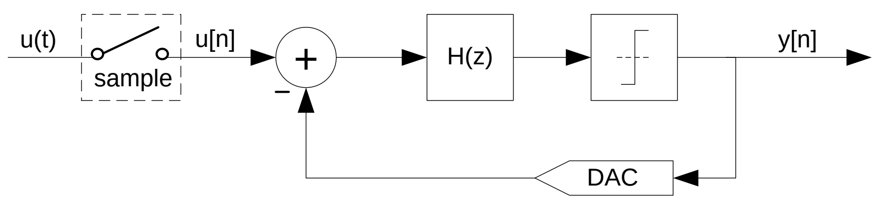


Figure 2.2: Structure of a first order $\Sigma\Delta$ A/D converter

The loop transfer function H(z) in figure 2.2 is a simple integrator expressed as $\frac{1}{z-1}$. Basically the loop filter can be implemented in the discrete time or continuous time domain. Continuous time sigma delta converters are discussed in section 2.1.4 in more detail.

2.1.1 Quantization

Quantization errors occur when input signals with infinite amplitude levels are mapped to a finite number of output values called quantization steps. In literature this quantization error is mostly described by a simplified noise model introduced by W. R. Bennett [5]. It approximates the real quantization error, the difference between input and quantized output, by a sawtooth function with amplitude E_0 and slope s. This is accurate enough for most applications. In figure 2.3 the sawtooth error function e(t), which is expressed with equation 2.1, is depicted. So the root mean square error can be calculated by equation 2.2.

$$e(t) = st, -\frac{E_0}{2s} < t < \frac{E_0}{2s} \tag{2.1}$$

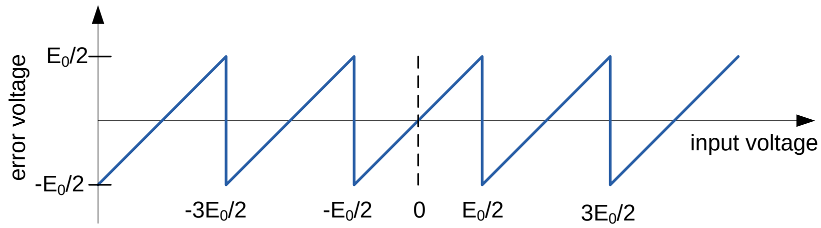


Figure 2.3: Quantization error signal approximated to a sawtooth function

$$\overline{e(t)^2} = \int_{\frac{-E_0}{2s}}^{\frac{E_0}{2s}} e(t)^2 \, \mathrm{d}t = \frac{{E_0}^2}{12}$$

$$e_{rms} = \frac{E_0}{\sqrt{12}} \tag{2.2}$$

With this equation in mind an approximation for the Signal to Quantization Noise Ratio (SQNR) of A/D converters can be expressed (2.3). As input signal the rms value of a full scale sine s(t) is assumed. The number of output bits is denoted by N.

$$s_{rms} = \frac{E_0 \cdot 2^{N-1}}{\sqrt{2}}$$

$$e_{rms} = \frac{E_0}{\sqrt{12}}$$

$$SQNR = 20 \cdot log_{10}\left(\frac{s_{rms}}{e_{rms}}\right) = 1.76\,dB + 6.02 \cdot N \tag{2.3}$$

Because signal and noise function have a different signal form the value of 1.76 dB comes up. In literature [16] the quantization noise is often assumed to be uncorrelated to the input signal and equally distributed over the whole spectrum. This is not always true. In [15] it was shown that for single tone inputs and sampling rates with an integer multiple of the signal frequency, the error is correlated to the input. This leads to an concentration of quantization noise around the harmonic frequencies of the input signal. Although the mean value of the noise keeps the same, this has an impact on other measured data like the Spurious-Free Dynamic Range (SFDR) or the Total Harmonic Distortion (THD). But for most application scenarios where the input is not a spectrally clear single tone but more or less a band of frequencies summed with noise the assumption of uncorrelated and randomly spread noise is valid.

2.1.2 Oversampling

In section 2.1.1 it was mentioned that quantization noise power can be assumed to be uniformly distributed over the whole spectrum for most cases. If the signal is oversampled, the quantization noise is scaled over a wider spectrum. Because the overall noise power has to stay the same, the noise per frequency unit decreases compared to Nyquist sampling. This fact is used to reduce the signal's in-band quantization noise. The higher the oversampling ratio the less noise is in the band of interest. Increasing the oversampling ratio M of a traditional A/D converter by a factor of four results in a SQNR change of +6 dB - or one bit - as equation 2.4 shows.

$$SQNR = 1.76\,dB + 6.02 \cdot N + 10\,log_{10}(M) \tag{2.4}$$

For sigma delta converters the achievable SQNR calculation from [16] is shown in equation 2.5. Here the number of loop filter stages L has a strong effect on the SQNR because of the noise shaping ability which is discussed in the next section in more detail. Furthermore only an oversampling ratio M greater than π shows an improvement compared to traditional oversampled A/D converters.

$$\begin{aligned} SQNR =& 1.76\,dB + 6.02 \cdot N + (2L+1) \cdot 10\,log_{10}(M) \\ &+ 10 \cdot log_{10}(2L+1) - (2L) \cdot 10\,log_{10}(\pi) \end{aligned} \tag{2.5}$$

2.1.3 Noise Shaping

The quantization process at the output of the converter is strongly non-linear. To analyze its influence on the system transfer function we linearize the quantizer by replacing it with an additive white noise source. This noise is independent of the input signal and adds noise so that the output corresponds exactly to the next quantization level. Considering the sigma delta converter depicted in figure 2.2, the output Y(z) can be written as described in equation 2.6 where $H(z) = \frac{1}{z-1}$ denotes the loop filter [11].

$$\begin{aligned} Y(z) &= \frac{H(z)}{1+H(z)} \cdot U(z) + \frac{1}{1+H(z)} \cdot E(z) \\ &= STF(z) \cdot U(z) + NTF(z) \cdot E(z) \end{aligned} \tag{2.6}$$

Noise transfer function and system transfer function derived from equation 2.6 result in equation 2.7.

$$\begin{aligned} NTF(z) &= 1 - z^{-1} \\ STF(z) &= z^{-1} \end{aligned} \tag{2.7}$$

It can be seen that the output is a delayed version of the input signal plus the high-pass filtered noise, hence the noise is shaped and moved to higher frequencies while the input signal stays unchanged.

For higher orders N the hardware effort scales linearly. The structure of a higher order ΣΔ A/D converter is shown in figure 2.4. The order of sigma delta modulators plays an important role when it comes to noise shaping. Figure 2.5 shows that higher order modulators shape the noise more efficiently to higher frequencies. For every doubling of the oversampling ratio the SQNR improves by 6 L + 3 dB. This results in lower sampling frequencies at the same resolution or in a higher resolutions at the same sampling frequency compared to lower order modulators [11].

Stability Issues: On the other hand stability issues rise for loop filter orders above two. Integrator overload is one of the main concerns. Thereby the output amplitude of the integrators next to the quantizer are overloading and the signal goes into saturation. This causes harmonic distortion in the output signal.

But also finding suitable poles and zeros for the transfer functions can become a sophisticated task. Further information on the topic of instability is found in [11].

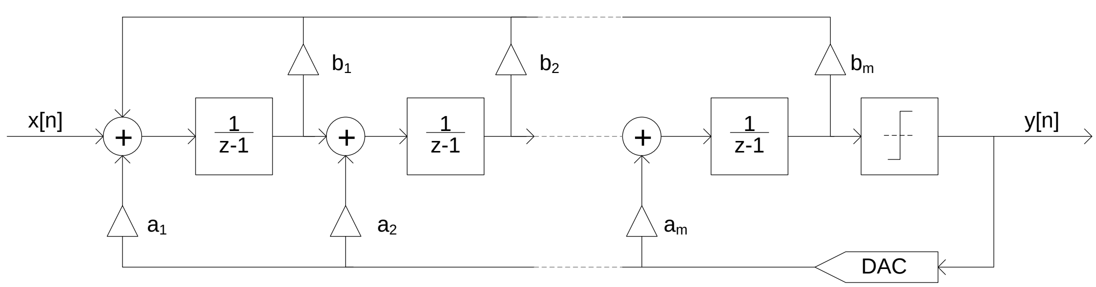


Figure 2.4: Structure of an M^{th} order ΣΔ A/D converter from [11]

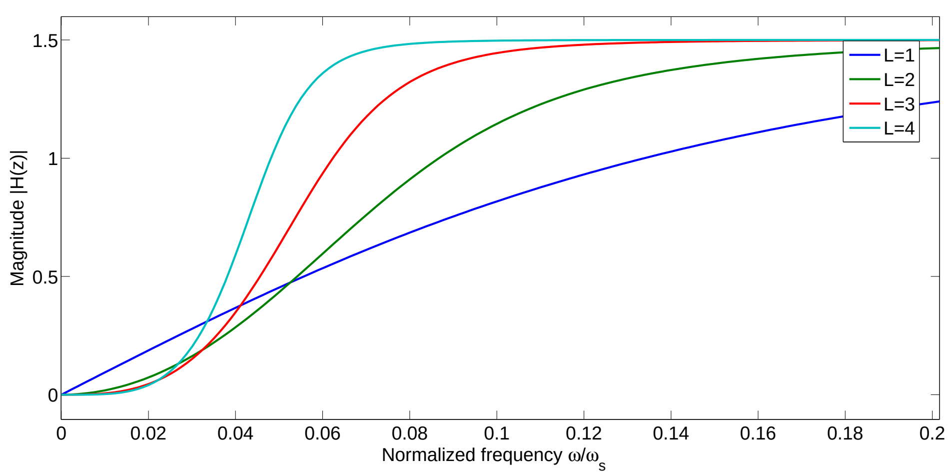


Figure 2.5: Noise shaping property dependent on number L of loop filter stages

2.1.4 Continuous vs. Discrete Time ΣΔ A/D Converters

In the previous sections it was mentioned that sigma delta modulators can be implemented with discrete time or continuous time loop filters. Now the implementation differences as well as advantages and trade offs are discussed.
Discrete time sigma delta modulators are build of loop filters which are often implemented in switched capacitor or switched current circuits [11]. The input signal is time discretized before it is integrated in the loop filters. Advantages of continuous time sigma delta modulators are:

- inherent anti aliasing filter
- higher sampling rates
- low power consumption
- no noise introduced by switched capacitors

One drawback of continuous time loop filters is their more sophisticated design.
Compared to discrete time sigma delta converters continuous time implementations are considered to be insensitive against aliasing [16]. This results from the fact that the signal sampling happens at the output of the loop filter. So the signal is first low-pass filtered before it is sampled and aliasing is introduced. This way, spectral components around multiples of the sampling frequency $M \cdot f_s$ are attenuated. After sampling, the aliased signal is fed into the quantizer and shaped by the feedback loop the same way quantization noise is processed. In contrast to discrete time modulators this anti aliasing filter comes with no additional effort. For higher order loop filters the anti aliasing performance rises because of the higher number of integrators [11].

Discrete time sigma delta converters sample the input signal in front of the loop filter. Therefore aliasing has already taken place when the loop filter processes the bit stream. That's why these modulators need additional analog anti-aliasing filters at their inputs. This contributes negatively to the implementation size and power consumption.

2.1.5 Digital Filtering

After converting the signal from analog to the digital domain the data is processed further by means of digital filtering. The goal of digital filtering is described by the following points:

- reduce oversampling
- prevent in-band aliasing introduced by decimation
- preserve the band of interest
- eliminate noise, especially the quantization noise of the A/D converter

The simplest decimation filtering structure would consist of a low-pass filter with subsequent decimation stage as shown in figure 2.6. It was proved in [20] and [25] that the same filter can be implemented more easily and with relaxed specifications in a multi stage filter approach. The main idea is to have multiple filters running at different sampling rates by factorizing the decimation rate R into m factors.

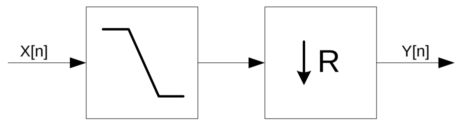


Figure 2.6: Basic decimation filter built of low-pass filter and decimation stage

$$R = R_1 \cdot R_2 \cdot R_3 \cdot \ldots \cdot R_m \tag{2.8}$$

Compared to a single filter design the advantage is a reduction of the sampling rate at an early stage in the filter chain. This has a positive effect on power consumption as the filters in the next stages are running at lower clock rates. Derived from equation 2.11 the power consumption can be assumed to reduce linearly with frequency. The following two sections give an overview of these practices.

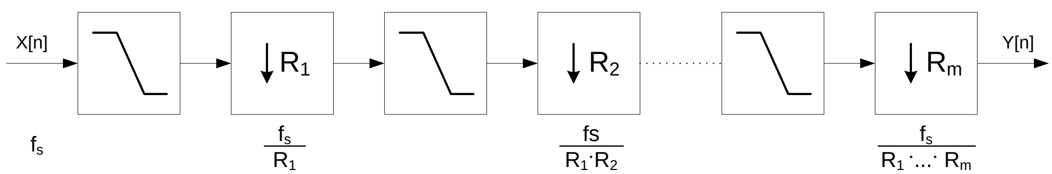


Figure 2.7: Multi stage decimation filter with reduction of sampling frequency f_s

Low-pass Filtering: $\Sigma\Delta$ A/D converted signals are highly oversampled and contain quantization noise due to the conversion from analog to digital signals. But fortunately the noise is shaped to higher frequencies outside the wanted signal bandwidth. The first step of the digital processing has to deal with low-pass filtering. This filter, often called *anti-aliasing filter* in literature, attenuates spectral components situated around the decimated sampling frequency f_s. These frequencies would fold down into the band of interest f_{bw} as a result of decimation. Therefore it is crucial to suppress spectral components around $f_s \pm f_{bw}$.

Decimation: After low-pass filtering the sampling rate can be adapted. This is done by dropping each R'th sample. In a hardware design, down sampling can be realized by clocking a register at lower clock rate. That way no additional hardware is required and the subsequent circuit is clocked with the decimated clock what in turn reduces power consumption.

2.2 CMOS Technology

Complementary Metal Oxide Semiconductor Technology (CMOS) is the most common process technology used for integrated digital circuits nowadays. It is known to be very power efficient. In the following section the influences on power consumption in CMOS networks are discussed.

2.2.1 Power Consumption

The total power consumption in CMOS processes is composed of static and dynamic power consumption as described in equation 2.9.

$$P_{tot} = P_S + P_D \tag{2.9}$$

Static power consumption is a consequence of current leakage in CMOS transistors. Although no switching activity is present a very small leakage current flows from V_{CC} to ground. This is expressed as the overall leakage current I_{leak} flowing into a circuit with supply voltage V_{CC}. The resulting dissipated power is described in equation 2.10.

$$P_S = V_{CC} \cdot I_{leak} \tag{2.10}$$

When it comes to dynamic power consumption we have to take several effects into account. One part is the switching current that has to load the output nodes of CMOS structures. This process is illustrated in figure 2.8. Another contribution to the dynamic power consumption is the short-circuit current I_{SC}. It originates from switching NMOS and PMOS transistors with finite slope at the same time. A short-circuit exists and current is flowing for a short period.

In CMOS technology the dynamic power consumption is calculated by equation 2.11. C_{pd} defines the power-dissipation capacitance, V_{CC} the voltage supply and f_0 the switching frequency. It can be seen that the supply voltage V_{CC} has the largest impact on the power consumption followed by the switching frequency.

$$\begin{aligned} P_D &= P_{switching} + P_{short-circuit} \\ P_D &= C_{pd} \cdot {V_{CC}}^2 \cdot f_0 + V_{CC} \cdot I_{sc} \cdot f_0 \end{aligned} \tag{2.11}$$

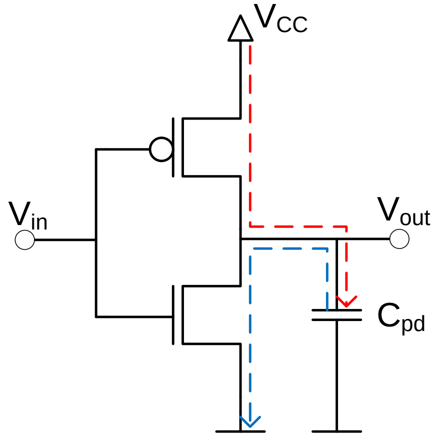


Figure 2.8: Charge transfer during switching activity in a CMOS equivalent circuit diagram

Chapter 3

Concept

3.1 Overview

This chapter gives an overview on the design process of the digital decimation filter. To get a deeper understanding of the overall system where the decimation filters should be applied, the receiver architecture is introduced. Afterwards the specifications of the filter are formulated. They depend on different properties of the whole signal processing chain and must be kept in mind during the design process.
Beside these specifications there are other objectives like chip area and power consumption that should be satisfied. The goal of this chapter is to find a concept for decimation filters which comply with the system requirements and at the same time have low power consumption and small size. Different implementations are summarized and their advantages and disadvantages are compared.

3.2 Architecture

Now the proposed receiver architecture for the wireless receiver is presented. The block diagrams in figure 3.1 and 3.2 show the main structure of the linear receiver. The concept involves one $\Sigma\Delta$ A/D converter quite early in the signal processing chain so that the vast majority of processing is done in the digital domain. First the analog and then the digital architecture is described in more detail.

Analog Processing: In the analog front-end the signal, received from the antenna, is filtered by a broadband SAW filter with high stopband attenuation. It belongs to the group of mechanical filters and is connected to the receiver externally. Then the signal is amplified by a Low Noise Amplifier (LNA) before it is down converted to an Intermediate Frequency (IF). To achieve a simple mixer structure and to avoid additional circuitry for image suppression, a Hartley image reject down converter is used. By mixing the RF signal with a 90° phase-delayed sinus, real and quadrature signals (I and Q) are generated. The polyphase filter does passband filtering on both of the two signals and introduces an additional 90° phase-shift on one of them. This results in an inverse image component which is completely extinguished when both signals are added together. Under the assumption that the circuit components have no tolerances this would be true. Under

real conditions the image is not perfectly suppressed.
The next stage in the analog domain is an Automatic Gain Control (AGC) which controls the dynamic range of the A/D converter. By adjusting the signal strength, the input of the sigma delta converter should be kept in range. This is necessary to avoid clipping and nonlinear side effects in the sigma delta modulator when full scale signals are converted.
The last analog stage in the architecture is the continuous time $\Sigma\Delta$ A/D converter. With its inherent anti-aliasing filter property, discussed in section 2.1.4, further analog filtering can be neglected at this stage.

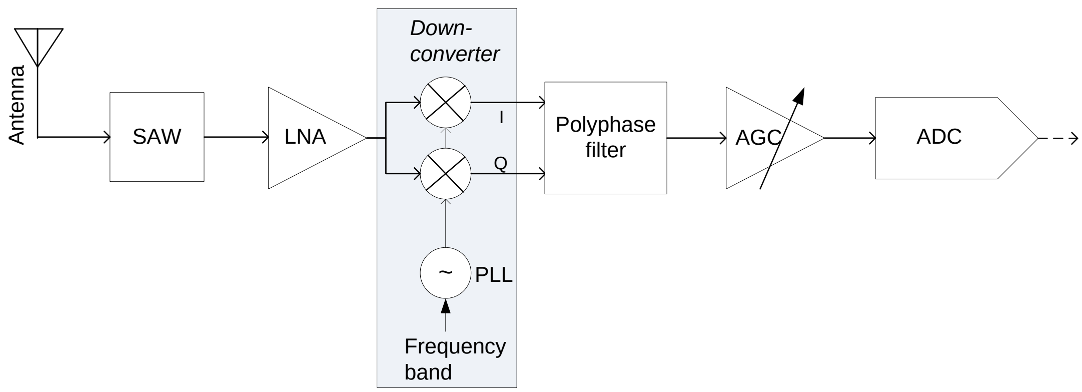


Figure 3.1: Analog front-end of the wireless receiver

Digital Processing: The output of the $\Sigma\Delta$ A/D converter is strongly oversampled and quantization noise is introduced during conversion. Therefore the first digital filter's purpose is down sampling and removing the shaped quantization noise. The further signal path depends on the receiver application. In the case of FSK or ASK modulation the signal is complex down-converted from its intermediate frequency to baseband. Then the real and quadrature path is channel filtered and further decimated. Afterwards FSK or ASK demodulation is done.
Another application of the receiver is Direct Sequence Spread Spectrum (DSSS) where the signal is processed in a separate path. This is an optional parallel signal processing unit that uses the decimation filter output as input signal. The whole digital signal path is illustrated in the block diagram in figure 3.2.

3.3 Specifications

A/D Converter: Figure 3.3 shows the system transfer function (STF) as well as the noise transfer function (NTF) of the continuous time sigma delta converter which was designed for use in this linear receiver. It can be observed, that through noise shaping the band of interest, which ranges from 250 kHz to 750 kHz, has very low noise components. Even the DSSS application that uses frequencies up to 1.3 MHz has at least 60 dB noise attenuation. An optimization of the NTF was done by placing one zero around 700 kHz.

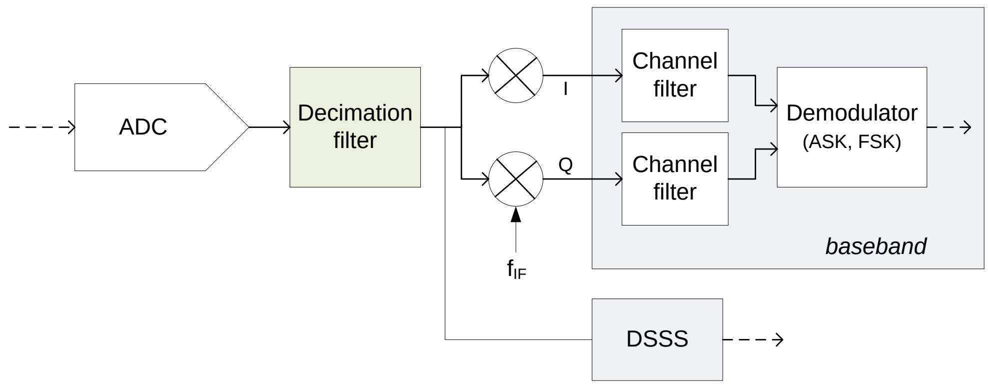


Figure 3.2: Digital signal processing in the wireless receiver

In this way the highest noise amount in the band of interest is additionally reduced.

The system transfer function depicts a characteristic low-pass curve. This is explained by the inherent anti-aliasing filter in continuous time sigma-delta converters. The cut-off frequency around 1.3 MHz is specified for DSSS applications.

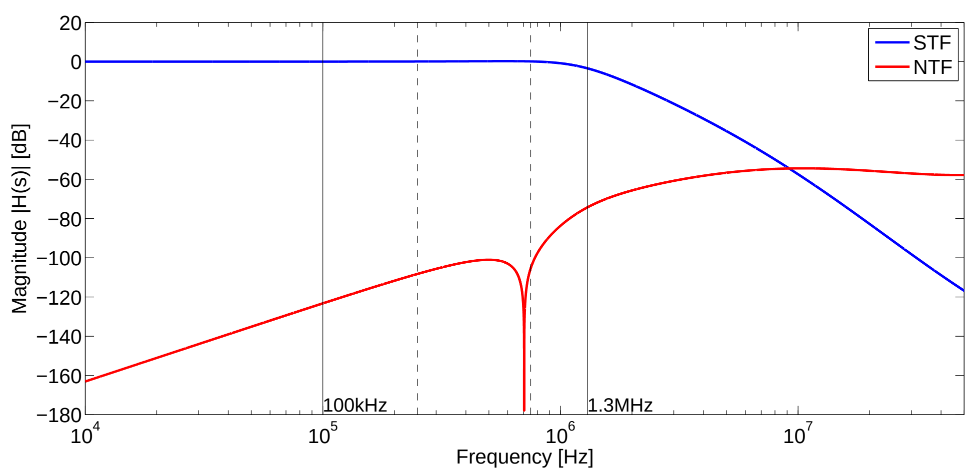


Figure 3.3: Sigma delta ADC system transfer function and noise transfer function

Frequency Band: The properties of $\Sigma\Delta$ A/D converters have been discussed in detail in section 2.1. With this knowledge we state the requirements of the decimation filters. In figure 3.4 the frequency specification of the digital receiver is depicted. This implies, that the signal has already been down-converted from radio frequency to the intermediate frequency (IF), or channel center frequency. The communication channel is placed within the channel band from 250 kHz to 750 kHz. The bandwidth should be variable between 10 kHz and 500 kHz for different applications. Optionally Direct Sequence Spread Spectrum should be implemented with the same architecture. This technique spreads the signal spectrum and therefore it is more robust against interferer. A bandwidth of 1.2 MHz is designated for DSSS mode ranging from 100 kHz up to 1.3 MHz. So the cut-off frequency of the low-pass filter in the decimation stage is specified with 1.3 MHz or higher to maintain frequencies required by DSSS. For the other case, where a channel is placed between 250 kHz and 750 kHz, the cut-off frequency could be even lower and consequently more quantization noise would be filtered. But this would require either one filter with adjustable cut-off frequency or two separate filters, one for each application. Two filters in parallel is no option when chip area should be kept small. Finding a filter structure in hardware that can be switched between two different states to satisfy different requirements is not trivial.

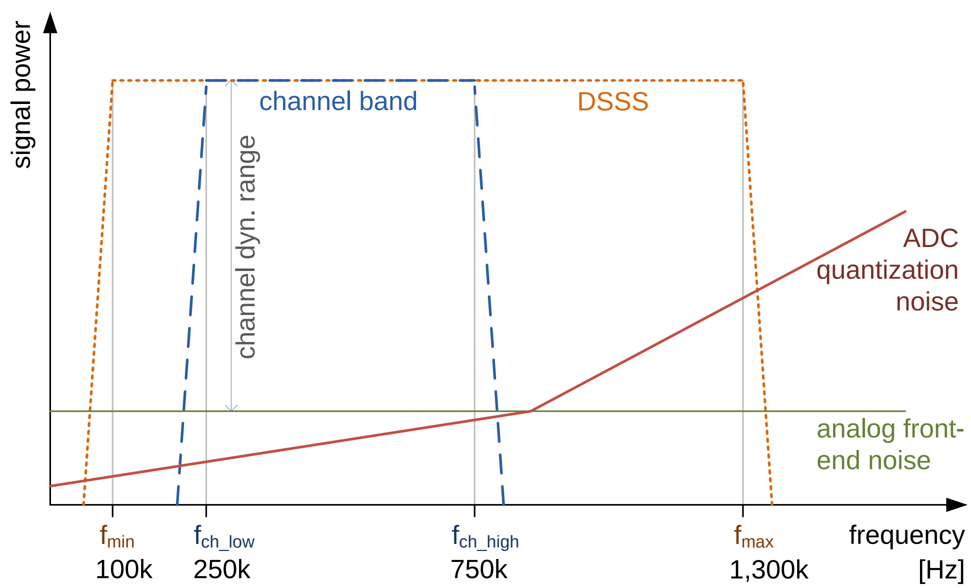


Figure 3.4: Frequency specification of the decimation filter

Decimation Factor: The decimation factor depends on two conditions. The first is the frequency specifications of the receiver defined in the previous paragraph. From sampling theory it is known that the minimum required sampling frequency is $2 \cdot f_{max}$, hence $2 \cdot 1.3\,MHz = 2.6\,MHz$. The sampling rate f_s of the A/D converter is 50 MHz, so the maximum possible decimation factor where the signal is not yet undersampled is calculated in equation 3.1.

$$R_{max} = \left\lfloor \frac{f_s}{f_{sig} \cdot 2} \right\rfloor = \left\lfloor \frac{50\,MHz}{2.6\,MHz} \right\rfloor = 19 \tag{3.1}$$

Second the decimation depends on constraints in the baseband down-converter. There an oversampling ratio of two is desired for better signal quality. As the channel band is specified from 250 kHz to 750 kHz, the highest possible frequency is 750 kHz with the Nyquist sampling rate f_{s_Nyq} of 1.5 MHz. To achieve an additional oversampling ratio of two the Nyquist frequency f_{s_Nyq} is doubled and results in the minimum sampling frequency f_{s_min}. Both, f_{s_min} and the resulting maximum decimation factor R_{max} are calculated by equation 3.2.

$$f_{s\_min} = f_{s\_Nyq} \cdot 2 = 3\,MHz$$

$$R_{max} = \left\lfloor \frac{f_s}{f_{s\_min}} \right\rfloor = \left\lfloor \frac{50\,MHz}{3\,MHz} \right\rfloor = 16 \tag{3.2}$$

Dynamic Range: Specifications on the system's dynamic range are shown in figure 3.5. This can be verified with the help of equation 2.5 where the achievable SQNR of $\Sigma\Delta$ A/D converters is calculated. In our case the sampling rate of the ADC is 50 MHz and the band of interest is specified with 1.3 MHz. This results in an oversampling ratio of approximately 19. With 4 bits at the ADC output and loop filter order of 4, the achievable SQNR results with equation 2.5 in 111 dB. This result reflects the dynamic range of the average quantization noise over the band of interest. When comparing this value with the noise transfer function in figure 3.3, one has to bare in mind that this is the average SQNR over the frequency band from zero to 1.3 MHz.
For input signals with maximum amplitude non-linearities occur within the A/D converter due to integrator overload. In the receiver architecture the Automatic Gain Control is responsible for adjusting the signal so that this is prevented. The Least Significant Bits (LSB) of the digitized signal contain the highest amount of noise.

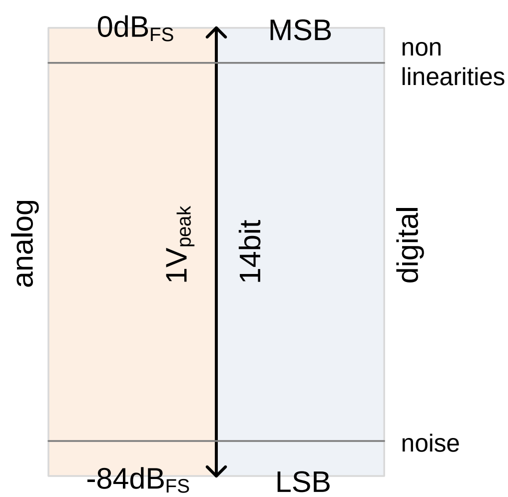


Figure 3.5: Dynamic range specification

3.4 Decimation Filters

With the system specifications in mind suitable decimation filters are investigated. The filter should be implemented in VHDL and is later applied to a CMOS process for a wireless receiver integrated circuit. This requires a small and power efficient hardware implementation. Most straight forward filter design methods usually result in very large filter structures with a huge amount of coefficients. In a hardware implementation, the number of coefficients is tried to be kept low because they consume a lot of space. They have to be stored in registers or implemented in combinatorial logic. The first case requires hardware multipliers whereas the combinatorial logic implements the multiplication inherently.

First of all simple FIR and IIR filters, created with the Matlab Signal Processing Toolbox, are investigated. This is a very simple and fast approach to get a feeling for the filter complexity. Criteria for the filter complexity are the filter specifications which are configured by the filter designer:

- filter bandwidth
- transition bandwidth
- stop-band attenuation
- pass-band ripple

If the filter specifications are too tight, e.g. the filter bandwidth relative to the sampling frequency is very narrow, the stopband attenuation is too high or the passband ripple is too low, then the complexity as well as the filter order grows. So a trade off between suitable filter characteristics and complexity has to be found.

In the next step more sophisticated filter structures are investigated and multi stage approaches are introduced. The list of literature dealing with decimation filtering is quite long so the most promising techniques and tricks for sampling rate conversion are presented.

3.4.1 Conventional FIR and IIR Filter Design

Filters with Finite Impulse Response (FIR) have some advantages over Infinite Impulse Response (IIR) filters that are of great demand in some applications. On the other hand IIR filters can be built with much smaller order than FIR filters to satisfy the same frequency specifications. Table 3.1 gives an overview on the main differences between FIR and IIR filters. When designing a filter, the trade offs always have to be kept in mind.

3.4.1.1 FIR Filter

In literature [18] different methods for designing FIR filters are introduced, e.g. window design, Parks-McClellan algorithm, frequency sampling and weighted least squares. The most popular method probably is the algorithm proposed by Thomas Parks and James McClellan in 1972 [21]. It is a variation of the Remez exchange algorithm and makes use of the Chebyshev approximation theory. Most software packages available nowadays implement this algorithm.

Property	FIR	IIR
number of coefficients	high	low
stable	yes	not inherently
phase	linear[1]	non-linear
limit cycles	no	yes
quantization influences	low	high

[1]for symmetrical coefficients

Table 3.1: Comparison between FIR and IIR filter properties

The Parks-McClellan algorithm is used to estimate the order of a FIR equiripple low-pass filter that satisfies the system's specifications mentioned in section 3.3. The following design parameters are considered:

- Low-pass filter
- Sampling frequency: 50 MHz
- Cut-off frequency: 1.3 MHz
- Pass-band ripple: 0.5 dB
- Stop-band attenuation: 60 dB
- Transition band: 525 kHz
- *Decimation factor: 16*

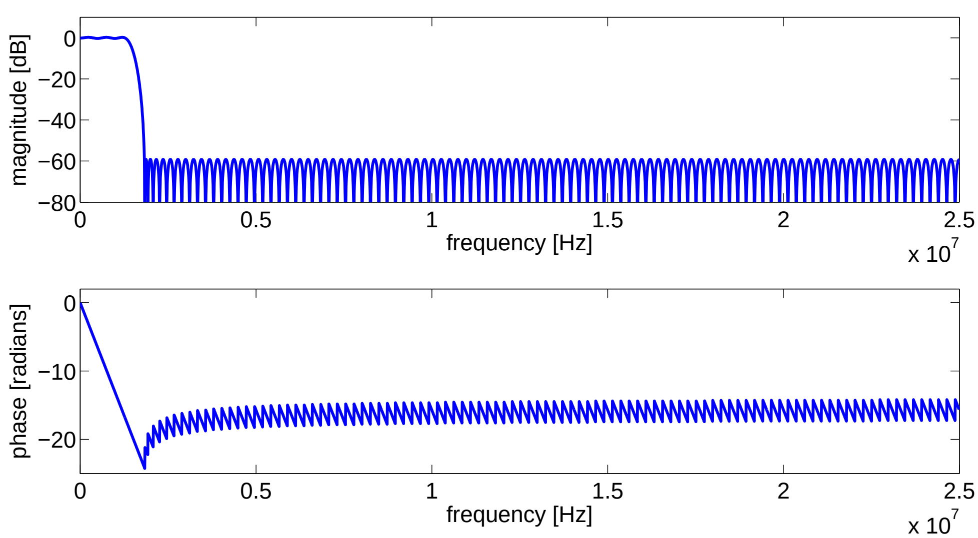


Figure 3.6: Frequency and phase response of the FIR equiripple filter

The transition band has been calculated to fit exactly the decimation factor. After decimation the Nyquist frequency is defined by equation 3.3. The transition bandwidth results from the fact that no aliasing should be introduced in the band of interest below 1.3 MHz. Equation 3.4 represents the frequency band which is outside the band of interest, where aliasing has no influence. To get the transition bandwidth of the filter this band has to be doubled and results in 525 kHz.

$$\frac{F_s/2}{16} = 1.5625\,MHz \tag{3.3}$$

$$1.5625\,MHz - 1.3\,MHz = 0.2625\,MHz \tag{3.4}$$

Filter Order: The result is a filter of order 210 that requires the same amount of coefficients to fulfill the specifications. To attain less pass-band ripple or even more stop-band attenuation the order would be even higher.

Group Delay: The filter has linear phase which means that the group delay is constant over the whole pass-band and no distortion is introduced. The constant delay is related to the order of the filter.

Decimation Factor: Rate decimation with a factor of 16 can be done after filtering without introducing aliasing to the band of interest. Therefore the transition band has been kept small and the transition center frequency f_{tc} has been chosen so that the new sampling frequency F_{s_new} (after decimation of 16) fulfills the condition in equation 3.5.

$$F_{s\_new} = 2 \cdot f_{tc} \tag{3.5}$$

Coefficient Quantization: When it comes to coefficient quantization the FIR transfer function is roughly the same for fixed point quantized coefficients with 12 bits as it is for a floating point implementation with 32 bit.

Remark 1 *Nevertheless a hardware implementation of this filter would be inefficient concerning area and power consumption. One has to take into consideration that the whole filter operates at the high frequency before decimation is done. From section 2.2 we know that with higher frequency comes higher power consumption. Summarized it can be said that this filter with linear phase suits the requirements but the very high order makes a low power hardware implementation not possible.*

3.4.1.2 IIR Filter

Infinite impulse response filters are in general designed by discretization of analog feedback filters which have well understood design methods. Bilinear transformation or the impulse invariance method are used for the transformation into the Z-domain. Examples of equivalent analog filters are: Elliptic, Chebyshev and Butterworth. Most software packages nowadays use optimized design methods that iteratively find the optimum filter

coefficients for a set of design parameters. Those parameters are e.g. cut-off frequency, transition bandwidth, filter order, pass-band, stop-band ripple and more.
For comparing different IIR design methods the specifications from section 3.3 are used. Again Matlab is used to calculate the filter coefficients. Figure 3.7 shows frequency response and group delay of various IIR filter implementations with the following specifications:

- Low-pass filter
- Sampling frequency: 50 MHz
- Cut-off frequency: 2 MHz
- Pass-band ripple: 0.2 dB
- Stop-band attenuation: 60 dB
- Transition band: 1 MHz
- *Decimation factor: 11*

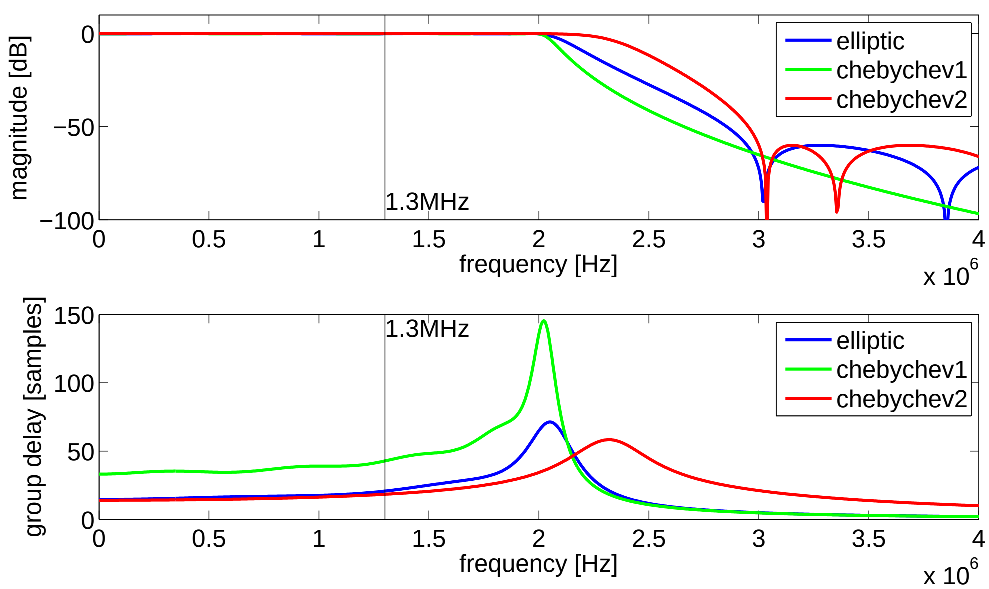


Figure 3.7: Frequency response and group delay of multiple IIR filter design methods

It has to be noted that these filter specification values vary from the values used for the FIR filter. The reason for this variation are the different filter properties especially the non-linear phase response of IIR filters that introduces distortions. For this reason the cut-off frequency is chosen higher so that the band of interest has *quasi* linear phase characteristics. This can be observed in figure 3.7. Consequently the decimation factor had to be reduced to a value of 11. Also the pass-band ripple was lowered in order to minimize the influence on the group delay. This adjustment had no impact on the filter order.

Filter Order: The resulting filter orders are varying from 6 to 21. The Elliptic filter implementation (blue) is of order six and requires 13 coefficients, the two Chebychev implementations (green and black) have order 10. A Butterworth design was also done but without satisfying success. Its filter order would have been 21 but the resulting filter coefficients in the Z-domain were too small for Matlab to produce a reasonable filter transfer function. Therefore it has been excluded from this comparison. From the filter orders it can be concluded that IIR filters in general satisfy the specification with much less coefficients compared to FIR filters.

Group Delay: Because of distortions, introduced by non-linear group delay, the cut-off frequency was shifted to higher frequencies. The purpose of this shift is to get an approximately linear phase in the band of interest from 100 kHz to 1.3 MHz. For this band the group delay amounts to a maximum of 12 samples difference between two frequency points.

Decimation: The maximum decimation rate had to be reduced for the IIR filter as a consequence of the shifted cut-off frequency. Otherwise aliasing would occur during decimation because of the wider pass-band that would fold into the signal band. Equation 3.6 and 3.7 clarify the decimation factor considerations. F_{s_new} is the sampling rate after decimation, f_{high} specifies the highest signal frequency, f_{st} the filter's stop-band frequency and R the decimation factor.

$$\begin{aligned} F_{s\_new} &= 2 \cdot \left(f_{high} + \frac{f_{st} - f_{high}}{2} \right) \\ &= 2 \cdot \left(1.3 + \frac{3 - 1.3}{2} \right) = 4.3 \quad [MHz] \end{aligned} \tag{3.6}$$

$$R = \left\lfloor \frac{F_s}{F_{s\_new}} \right\rfloor = \left\lfloor \frac{50\,MHz}{4.3\,MHz} \right\rfloor = 11 \tag{3.7}$$

Remark 2 *With an elliptic IIR filter it is not possible to decimate the signal with the maximum factor because either distortion would be introduced or aliasing would occur. Thus a higher cut-off frequency and lower decimation rate is unavoidable. The lack of noise filtering has to be compensated with the channel filter that follows the decimation stage. This may lead to more complex filter structures there. Because group delay is the limiting factor for rate decimation a phase equalization filter would be the key for even higher decimation factors. But it has to be investigated if the additional costs for an (approximated) linear phase IIR filter really pays off compared to a smaller filter with the subsequent digital receiver blocks running on a higher sampling rate.*

3.4.2 Cascaded Integrator Comb Decimation Filter

The last section about FIR and IIR filters demonstrated that a straight forward implementation of these filters is not appropriate for our requirements, therefore alternative filter structures particularly suitable for rate decimation are investigated.

3.4.2.1 Characteristics

The Cascaded Integrator Comb (CIC) filter was introduced by Eugene B. Hogenauer in 1981 [10]. Since then, it has become very popular in digital signal processing applications because of its simple and efficient implementation. It consists of integrators and comb sections only. By swapping these components, the filter can be used for rate decimation or interpolation. The main advantages of CIC filters are summarized in the following list:

- decimation factor can be changed without altering the filter structure
- multiplier free design
- no need for coefficient storage
- half the filter runs at the lower sampling rate
- finite impulse response
- linear phase

Disadvantages of CIC filter implementations are:

- large register bit-widths for high rate change factors or filter orders
- limited parameters to change the filter characteristics
- rather large passband droop for higher decimation factors and filter orders

The CIC filter consists of integrator and comb sections in a cascaded form. The system transfer function is shown in equation 3.8 where $H_I(z)$ is the integrator's and $H_C(z)$ the comb's transfer function, R the rate decimation, N the order and M the differential delay.

$$H_I(z) = \frac{1}{1 - z^{-1}}, \qquad H_C(z) = 1 - z^{-RM}$$
$$H(z) = H_I^N(z) \cdot H_C^N(z) = \frac{(1 - z^{-RM})^N}{(1 - z^{-1})^N} \tag{3.8}$$

The implementation of the filter is shown in figure 3.8. It can be seen that N integrators are followed by the same amount of differentiators with a rate decimation block in between.

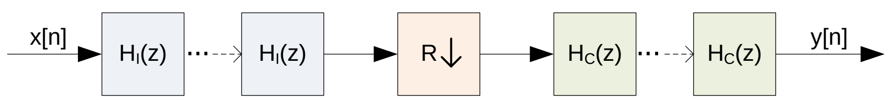
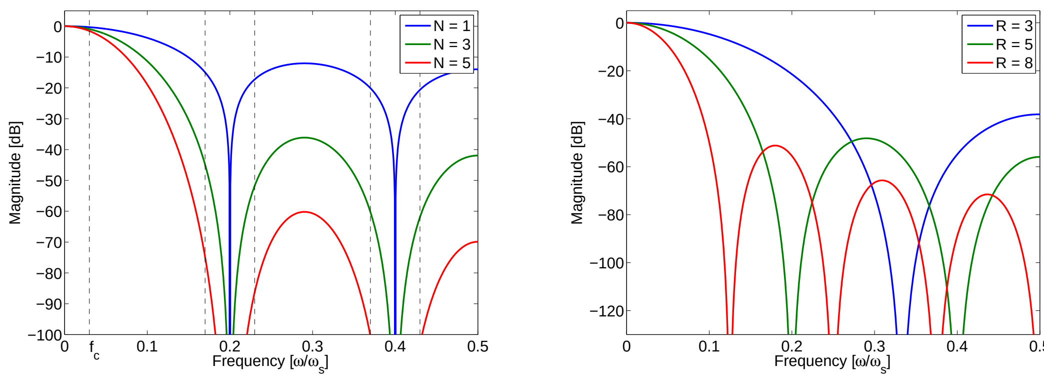


Figure 3.8: N^{th} order recursive CIC filter structure

One parameter of the filter is the differential delay M. It controls the differentiation stage only and introduces additional zeros in the frequency response for higher values. In general this parameter is set to one or two.

By increasing the filter order the number of integrators and comb sections is also equally increased. With higher order comes higher aliasing rejection at the cost of increased passband droop. The same side effect of passband droop is valid for the rate decimation parameter. To illustrate this behavior, figure 3.9(a) shows the transfer function of CIC filters with different orders and figure 3.9(b) shows transfer functions for various decimation factors.



(a) Various CIC orders (R=5, M=1) (b) Various CIC decimation rates (N=5, M=1)

Figure 3.9: Frequency response of a CIC filter with variation of order N and decimation rate R

Aliasing: The problem of aliasing, which is introduced by rate decimation, is also visualized in figure 3.9(a). The signal band ranges from zero to the cut-off frequency f_c. Those frequency bands that will fold down into the signal band are located at the zeros of the transfer function. They are marked with vertical dashed lines. Although the zeros heavily attenuate these bands and help minimizing aliasing there are always some frequency components suppressed less efficiently. The frequency component with the highest aliasing contribution is mostly situated where the first aliasing band cuts the main lobe of the transfer function.

Filter Order: In can be seen that higher filter orders attenuate the aliasing bands better than lower orders because they have much steeper transition bands. As a drawback, higher orders have lower passband cut-off frequencies. Consequently this leads to higher passband droop.

Decimation Factor: Figure 3.9(b) illustrates frequency responses of CIC filters with various decimation factors. It can be seen that for higher decimation rates the filter's passband gets narrower. The discussed effect of passband droop appears if the signal band is not scaled with the decimation factor.

In [10] two tables can be found that help choosing the filter parameters. One table lists passband attenuation values for different signal bandwidths and filter orders. The other table is listing aliasing attenuation values for certain filter orders under respect of signal bandwidth.

Gain: Due to their composition of integrators, CIC filters do have strong signal gain. The maximum gain can be calculated by equation 3.9, introduced by [10].

$$G_{max} = (R \cdot M)^N \tag{3.9}$$

The maximum register growth is then calculated by formula 3.10 [10] with B_{in} defining the number of input bits and B_{max} being the Most Significant Bit (MSB).

$$B_{max} = \lceil N \cdot log_2(R \cdot M) + B_{in} - 1 \rceil \tag{3.10}$$

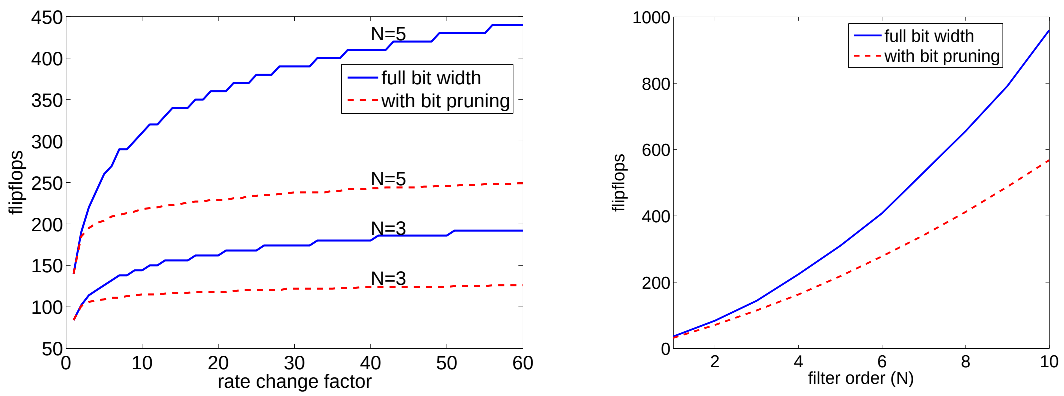
Bit Pruning: With this maximum number of bits at each stage the signal can be represented without overflows because B_{max} is an upper bound for register growth. This means that the number of bits necessary at each stage can be further reduced. Therefore an appropriate mechanism has been introduced in [10]. The main idea behind this theory is that the process of quantizing the signal at the output stage from B_{max} bits to the desired output length introduces the same amount of quantization error as if the signal bit-width was pruned already inside the 2N filter stages. There exists a total of $2\,\mathrm{N}+1$ error sources introduced by bit pruning. 2 N sources occur at the input of the stages and one error appears when truncating or rounding the last stage's result to fit into the output register. A distinction between rounding and truncation is not necessary because it was shown by the author that the error statistics of both operations are the same except for the first stage and the output register. In most implementations full precision is used in the first stage and rounding is done at the output.

An example should illustrate the bit pruning method. A filter with order 5, rate change factor 10 and differential delay 1 is assumed. Input and output registers should be 14 bits wide. The determination of the maximum bit-width B_{max} is done by equation 3.10. The pruned bit-width vector B_{pruned} is calculated with the help of Hogenauer's formulas described in [10].

$$\begin{aligned} B_{max} &= \lceil 5 \cdot log_2(10 \cdot 1) + 14 \rceil = 31 \\ B_{pruned} &= [31 \quad 27 \quad 25 \quad 23 \quad 21 \quad 20 \quad 19 \quad 18 \quad 17 \quad 17] \end{aligned} \tag{3.11}$$

In equation 3.11 the effective number of bits saved per stage, when applying the pruning method, is shown. The vector B_{pruned} represents the bit count from the first integrator stage to the last comb stage (2 N). All together the bit count was reduced from $B_{max} \cdot N = 310$ to 218 bits. This means a saving of 92 bits. The example also illustrates that the first stage was not truncated and it can be observed that the bit-width is reducing from the first to the last stage.

For different rate change factors the flipflop effort is shown in figure 3.10(a). Input and output registers are 14 bits wide. It can be seen that the bit pruning method has an remarkable impact on the size of a CIC filter. In figure 3.10(b) it is shown that with higher filter orders also the profit of this method grows.



(a) Bit count as a function of rate change factor (b) Bit count as a function of filter order

Figure 3.10: Bit count comparison of implementations with and without pruning method

3.4.2.2 Non-Recursive Structure

In the previous section the recursive CIC structure with integrator and comp sections has been introduced. Now the system function from equation 3.8 is expressed as geometric series to get the non-recursive structure representation shown in equation 3.12.

$$H(z) = \left[\sum_{k=0}^{RM-1} z^{-k}\right]^N = (1 + z^{-1} + z^{-2} + \ldots + z^{-RM+1})^N \tag{3.12}$$

This implementation has no more feedback loops and therefore minimizes register growth and the need for very large registers. By factorizing equation 3.12 we get the expression in 3.13. The decimation factor has to be an integer power of two $R = 2^K$ with a positive integer K.

$$H(z) = (1 + z^{-1})^N \cdot (1 + z^{-2})^N \cdot (1 + z^{-4})^N \ldots (1 + z^{-2^{K-1}})^N \tag{3.13}$$

This factorization allows us to implement the filter with K stages, each having a rate decimation factor of two. Compared to the recursive implementation this form has multiple rate decimations along the signal path and not only one centered in the middle of the filter. In this way the sampling rate is constantly reduced throughout the signal path. In [18] it was mentioned that **the power consumption for this implementation is lower for filter orders greater than three and rate change factors above eight.** Figure 3.11 shows the non-recursive implementation for a N^{th} order CIC filter with rate decimation factor of eight and a differential delay of one. The decimation factor R can be represented as $8 = 2^3$, so three structures are necessary for the implementation.

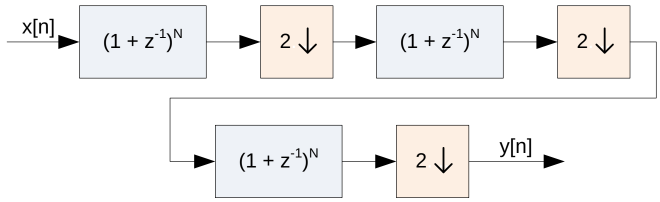


Figure 3.11: N^{th} order non-recursive CIC filter with rate decimation factor R=8

When the transfer functions of these stages are investigated it can be seen that they are all the same. The higher order polynomials in the second and third stage have been reduced by the rate decimation stage before. Therefore it is essential for the decimation factor to be an integer power of two, otherwise this advantageous factorization can not be performed in this way. Another approach based on the factorization of the decimation factor into prime numbers may also be applied. This method is explained in [18]. Consequently the decimation factor can be chosen arbitrarily and the total non-recursive filter is composed of stages where the respective rate change is equal to the prime factors.

Polyphase Decomposition: The next step toward a more efficient implementation is the so called polyphase decomposition. This method helps reducing sampling rate at an early stage in the signal path. Thereby a transfer function is decomposed into two separate functions that can be evaluated in parallel because they are completely decoupled from each other. Afterwards they are added again to get the overall result. Applied to each non-recursive stage, this would look like depicted in figure 3.12 for a 4^{th} order CIC filter. The equivalent transfer function is derived in equation 3.14.

$$\begin{aligned} H(z) &= (1 + z^{-1})^4 = 1 + 4z^{-1} + 6z^{-2} + 4z^{-3} + z^{-4} \\ &= (1 + 6z^{-2} + z^{-4}) + (4 + 4z^{-2}) \cdot z^{-1} \\ &= H_A(z) + H_B(z) \cdot z^{-1} \end{aligned} \tag{3.14}$$

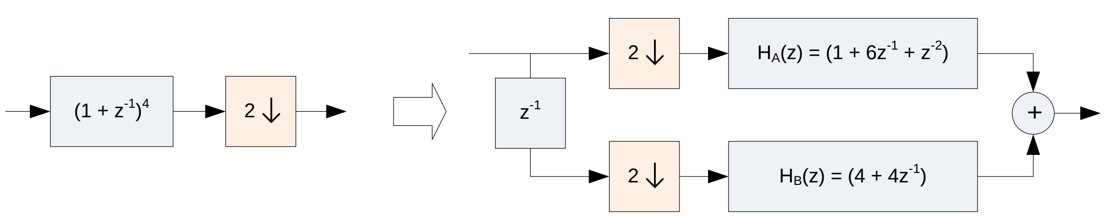


Figure 3.12: Polyphase decomposition of one non-recursive CIC stage with order N=4

Implementation Issues: The result of factorization and polyphase decomposition has following advantages and trade offs. Obviously the advantage of this structure is the rate decimation at a very early stage of the structure so that all calculations are done at the low sampling rate. The register effort of the polyphase decomposed structure keeps the same, they are just allocated to different processing paths. But there is a difference in bit growth between these two implementations. **The non-recursive implementation needs only N additional bits per stage** according to [18]. So the first stage's output has $B_{in} + N$ bits, the second stage $B_{in} + 2N$ and the i^{th} stage $B_{in} + iN$. In contrast the recursive structure needs B_{max} bits for each stage in the worst case.

One may notice that coefficients have been introduced with the non-recursive structure. This is true, but as the coefficients are integer numbers they are quite easy to factorize and implement as shift and add operations. For example the coefficients from $H_A(z)$ can be represented as shown in equation 3.15 where x is the input stream and $x << \alpha$ denotes a left shift of x by α bits.

$$\begin{aligned} 4 \cdot x &= (x << 2) \\ 6 \cdot x &= (x << 2) + (x << 1) \end{aligned} \tag{3.15}$$

3.4.2.3 Recursive vs. Non-Recursive

To be able to extract advantages and disadvantages of these two implementations a comparison of size and power consumption has been done. Because of its structure the non-recursive implementation can only handle decimation factors that are integer powers of two. Hence rate change factors 2,4,8 and 16 are investigated. Furthermore only registers are considered in the calculations. This simplification can be done because the number of used adders is roughly equivalent to the number of registers. For the non-recursive structure the number of adders can vary with the coefficients of the polyphase decomposed function that in turn depends on the filter order.

Area Usage: Section 3.4.2.1 already discussed the bit pruning method that calculates the minimum number of bits required for each section of the recursive implementation. This method is used for the area estimation of the recursive structure. Equation 3.16 shows the formula for area determination of the non-recursive implementation with rate change factor R, filter order N and B_{in} input bits. The resulting bit count A reflects the required flip-flops in a hardware design.

$$A = ld(R) \cdot N \cdot B_{in} + \sum_{i=0}^{ld(R)-1} iN^2 \tag{3.16}$$

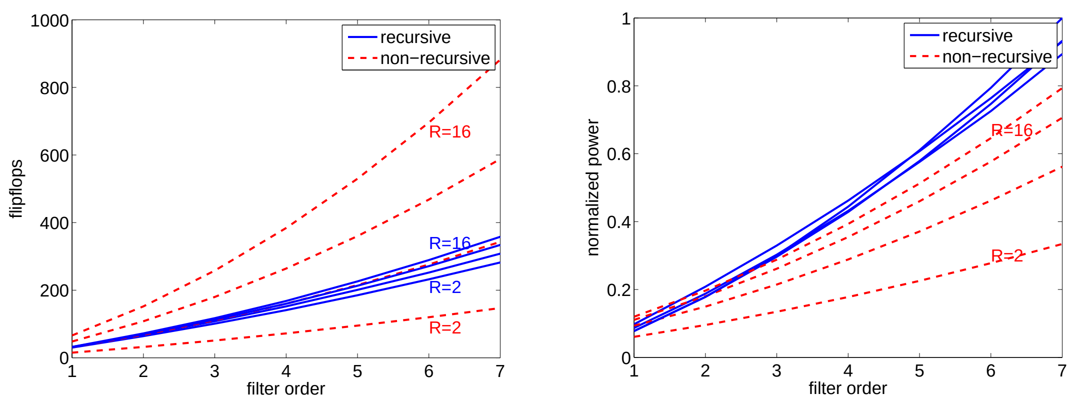
Power Consumption: Again the power consumption is estimated with the help of registers only because the number of adders is approximately equal. For this estimation each register is weighted with its sampling rate. The power consumption reduces linearly with the sampling frequency as mentioned in section 2.2. The resulting normalized power P for a recursive implementation is calculated by equation 3.17 where b_i is the bit-width of the i^{th} stage.

$$P = \sum_{i=1}^{N} b_i + \sum_{i=N+1}^{2N} \frac{b_i}{R} \tag{3.17}$$

For the non-recursive structure the estimated power consumption is calculated as described in equation 3.18. It consists of one register running on the highest sampling rate and a sum of registers running on decimated rates. The last term represents one register that has to be subtracted from the last stage for correction of the result.

$$P = B_{in} + \sum_{i=1}^{ld(R)} (iN^2 + NB_{in}) \cdot \frac{1}{2^i} - (B_{in} + N \cdot ld(R)) \cdot \frac{1}{R} \tag{3.18}$$

Comparison: Now the two implementations are compared with each other. For the recursive structure the bit pruning method introduced in section 3.4.2.1 is used to calculate the register bit-width of each stage. Both input and output registers are chosen to have a bit-width of 14.



(a) Flipflop usage

(b) Power consumption

Figure 3.13: Comparison of recursive and non-recursive CIC filter with R=2,4,8,16

Figure 3.13(a) compares the area usage as a function of filter order. The decimation factors 2,4,8 and 16 are printed as separate curves. It can be concluded from this diagram that in general the non-recursive implementation needs more flip-flops when higher decimation rates are used. Considering the quite narrow distances between the curves of the recursive implementation it can be concluded that for even higher decimation rates, the size will not increase significantly.

Although it was shown that the non-recursive structure consumes more area the diagram in figure 3.13(b) presents the advantages of the non-recursive implementation where significantly less power is consumed especially for lower decimation rates and higher filter orders.

Another remarkable point is that the power consumption curves of the recursive structure are close together. This means that the decimation rate in general has no major influence on power consumption for this structure. This behavior can be argued with the very small increase of register bit-width for higher rate change factors and the consequently lower power consumption of the comb sections.

3.4.2.4 CIC Sharpening

This section deals with a method for compensating a CIC filter's passband droop. Chapter 3.4.2.1 discussed this drawback when using CIC filters in more detail. It was shown that the passband droop increases with higher orders and larger rate change factors. In literature various methods for compensation were published. One of the first proposals has been published by Kaiser and Hamming [14]. They presented a sharpening method that uses the same CIC filter multiple times. Equation 3.19 gives the overall transfer function of the structure where H(z) is the transfer function of the CIC filter.

$$H_{sh}(z) = 3\,H^2(z) \;-\; 2\,H^3(z) \tag{3.19}$$

A hardware implementation of this filter gets very large quite quickly. So an improved implementation has been presented in [12]. Although this method is derived from equation 3.19, the transfer function is modified to need only half the coefficients. Furthermore the sharpening is done at half of the input sampling rate and therefore is more efficient concerning power consumption.

A new approach for sharpening the passband of CIC filters with less hardware effort was proposed in [6]. It is based on a separate filter with cosine frequency response described in equation 3.20 where R denotes the rate change factor.

$$H(e^{jR\omega}) = 2\,b \cdot cos(R\omega) + a \tag{3.20}$$

The equivalent transfer function in the Z-domain is described in equation 3.21 with coefficients $a = 72 \cdot 2^{-6}$ and $b = -4 \cdot 2^{-6}$ [6].

$$\begin{aligned} H(z^R) &= b\,z^{-R} + a\,z^{-2R} + b\,z^{-3R} \\ H(z^R) &= -2^{-4} \cdot (z^{-R} - (2^4 + 2)z^{-2R} + z^{-3R}) \end{aligned} \tag{3.21}$$

This FIR filter has three coefficients which are easy to implement with low hardware costs. Shift and add operations can be used explicitly. One major advantage is that it can be implemented completely at the lower sampling rate as illustrated in figure 3.14.

The implementation of the CIC filter itself is independent of the sharpening structure. For filter orders lower than four, the sharpening stage has to be implemented as a cascade of N stages where N is the CIC filter's order. For higher orders, N-1 stages can be used in cascade. Figure 3.15 depicts the frequency response of the sharpening filter. It can

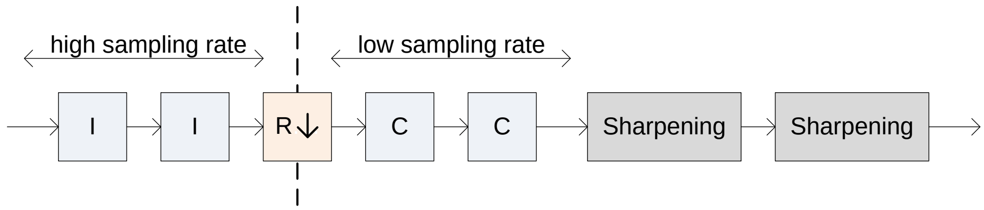


Figure 3.14: Sharpened CIC filter with order N=2

be seen that higher orders have more passband compensation but at the same time more side lobe gain is introduced. As a consequence this also influences the attenuation of the aliasing bands which are located around the filter's zeros.

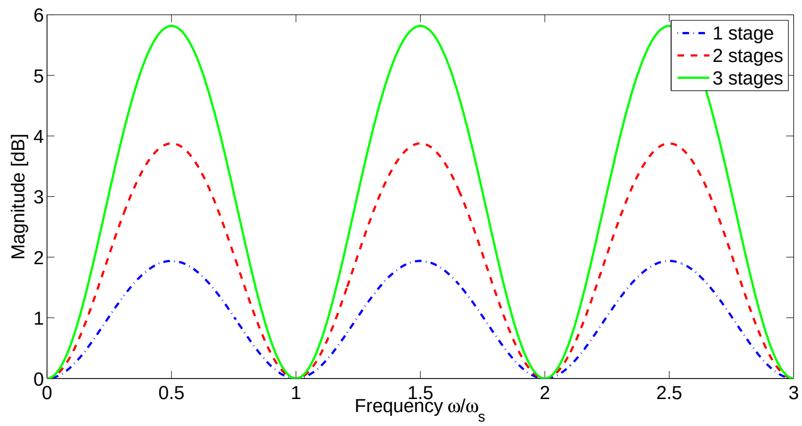


Figure 3.15: Frequency response of sharpening filter

Block diagram 3.16 illustrates the implementation of one sharpening stage in hardware. Three registers, three adders and one inverter are used for one stage. The bit-width of these hardware blocks does not increase for cascaded stages. The need for additional bits to improve the overall SNR has to be analyzed in the final design.

When several stages are cascaded there is a way of saving one register per stage by leaving away registers that connect two structures. Figure 3.17 illustrates this process. Of course this is only possible if timing constraints are not violated. But as mentioned before, the sharpening structures run on the lower frequency. This implicates that timing constraints are more relaxed.

Example: A 5^{th} order CIC filter with rate decimation factor of five and differential delay one should be compensated with the method described above. The required number of cascaded sharpening stages was said to be N-1 because the order of the CIC filter is higher than three. Figure 3.18(b) shows the transfer function of the genuine CIC filter, the compensation filter and both combined together. The shaping of the function can be recognized. The improvement in the passband is illustrated in figure 3.18(a) more clearly.

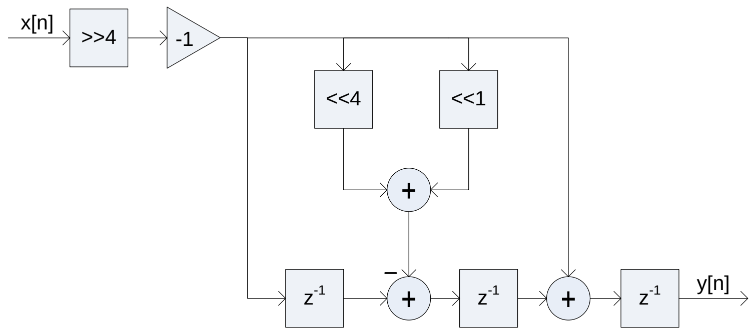


Figure 3.16: Hardware implementation of one sharpening stage

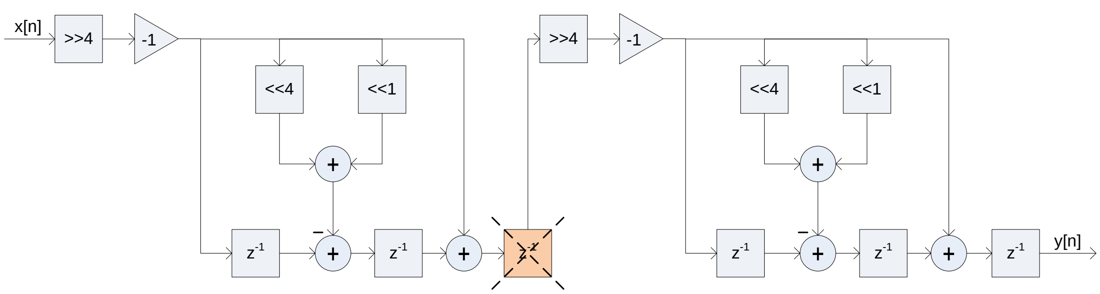
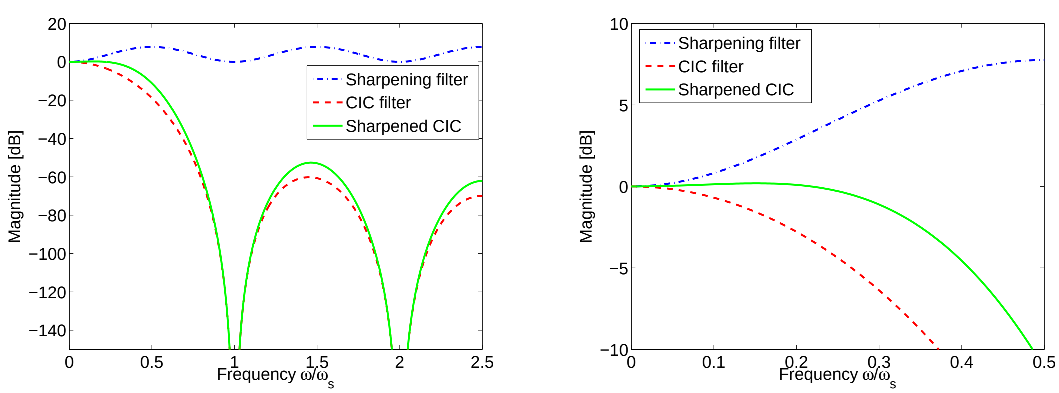


Figure 3.17: Register saving for implementations with multiple sharpening stages



(a) Overall transfer function

(b) Improved passband

Figure 3.18: CIC sharpening filter: N = 5, R = 5, M = 1

3.4.3 Half-band Filters

Half-band filters are low-pass filters with favorable properties. They have a symmetrical frequency response around $\omega = \pi/4$ where the transition region is centered. That's the reason why half-band filters are often used in multi-rate systems where rate decimation by a factor of two is done. Polyphase decomposition is the preferred choice in a hardware implementation because almost the whole filter structure can operate on the lower sampling rate.

Usually filters are specified by the following parameters: pass-band frequency f_{pass}, stop-band frequency f_{stop}, pass-band ripple δ_p and stop-band ripple δ_s. There were three conditions mentioned in [13] that must be fulfilled when designing half-band filters:

1. frequency symmetry: $f_{pass} + f_{stop} = \frac{f_s}{2}$
2. magnitude: $\delta_p = \delta_s$ for FIR and $\delta_p = \frac{1}{4} \cdot \delta_s^2$ for IIR filters
3. filter order: must be an odd integer

For FIR half-band filters with linear phase an additional condition mentioned in [18] has to be fulfilled: the total number of filter taps plus one has to be an integer multiple of four. Otherwise the linear phase criterion is not met.

One additional advantage of half-band filters is that half of their coefficients are zero. This results in smaller hardware implementations because half of the multiplications can be avoided. The following paragraph describes the conditions of FIR half-band filters in more detail.

FIR Half-band Filters: The frequency response is symmetrical around $\pi/4$ so that $f_{pass} + f_{stop} = f_s/2$. Pass-band and stop-band ripple are equally high. The center of the transition band has -6 dB magnitude as indicated in figure 3.19 where the frequency response is shown.

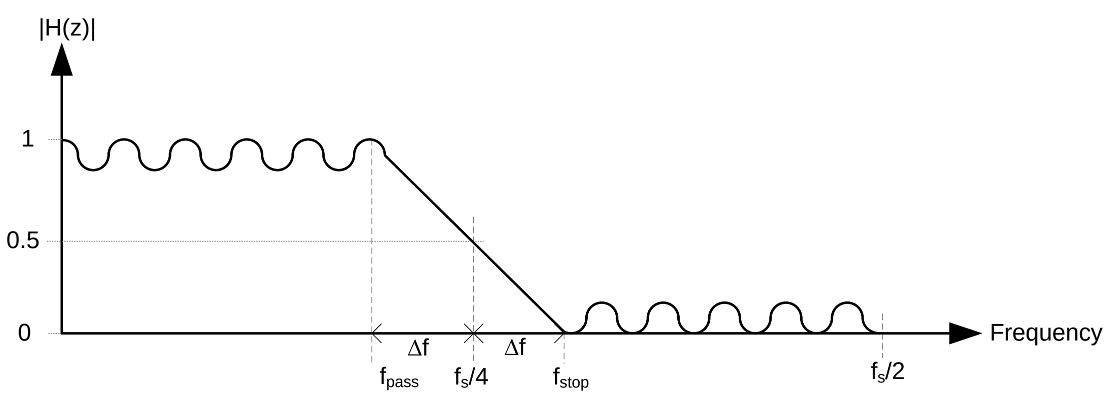


Figure 3.19: FIR half-band filter frequency response

The frequency response is described by

$$H(z) = \sum_{n=0}^{2M} h[n] \cdot z^{-n},$$

with M being an odd number denoting the amount of filter taps h. The center coefficient is

$$h[M] = 0.5$$

and each second coefficient is zero:

$$h[M + 2r] = 0 \quad for \quad r = \pm 1, \pm 2, ..., \pm \frac{M-1}{2}.$$

IIR Half-band Filters: In general IIR filters need less coefficients than FIR filters to fulfill the same specifications. This is also true for half-band filters. In [13] it was mentioned that the elliptic half-band filter with odd filter orders and implemented as two parallel all-pass networks results in the most efficient implementation. Compared to the FIR half-band filter the IIR filter has -3 dB magnitude at its transition center frequency and less passband ripple [13].

Cascaded Half-band Filters: To achieve higher decimation rates, several half-band filters are implemented in cascade. With this approach only rate change factors with a power of two are possible. Figure 3.20 shows the frequency response of up to four FIR half-band filters in cascade. They all have the same coefficients and reduce the sampling rate by a factor of two. Four stages thus have a rate decimation of 16. One half-band stage has filter order 38. The center coefficient is 0.5 as mentioned before, 20 coefficients are different from zero and 18 coefficients are exactly zero. Because symmetry around the center coefficient exists the phase response is linear.

When studying figure 3.20 more closely it can be seen that the transition slope of cascaded stages is steeper than the slope of a single stage. Although the filter specification stays the same for all filter stages the last stage operates at much lower sampling frequency and therefore shows a steeper transition when observed at the high sampling rate.
Compared to the single FIR filter example in section 3.4.1.1, the 4 stages half-band filter approach fulfills approximately the same requirements. Stop-band attenuation of 60 dB, passband ripple smaller than 0.5 dB, passband of 1.3 MHz and decimation factor of 16. Looking at the implementation details the cascaded half-band filter has several advantages over the single FIR filter. First it uses less coefficients because half of them are zero. So only $4 \cdot 20 = 80$ coefficients are needed. The single stage needs 210 coefficients for the same requirements. One might have noticed that the passband ripple was specified to be 0.5 dB in the single FIR example. The design criterion of FIR half-band filters says that passband and stop-band ripple have to be equal $\delta_p = \delta_s$. With a stop-band ripple of -60 dB the passband ripple is approximately 0.02 dB and therefore a lot smaller than the ripple in the single FIR example.

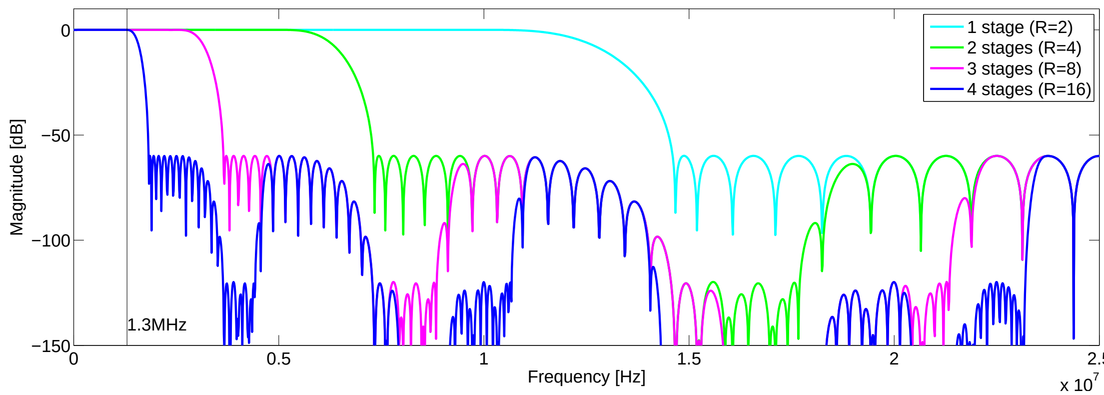


Figure 3.20: Frequency response of cascaded FIR half-band filters

Another major advantage is the stepwise rate decimation in a multi filter approach. By dividing the sampling rate at each stage, the power consumption is reduced because at each single stage the hardware performs operations only half of the time compared to the previous stage.

Figure 3.21 shows the same example as before but implemented as elliptic IIR half-band filter. It was already mentioned that the elliptic implementation is the most efficient one. The filter specifications are about the same, the cut-off frequency should be located at 1.3 MHz and the stop-band attenuation should reach 60 dB.

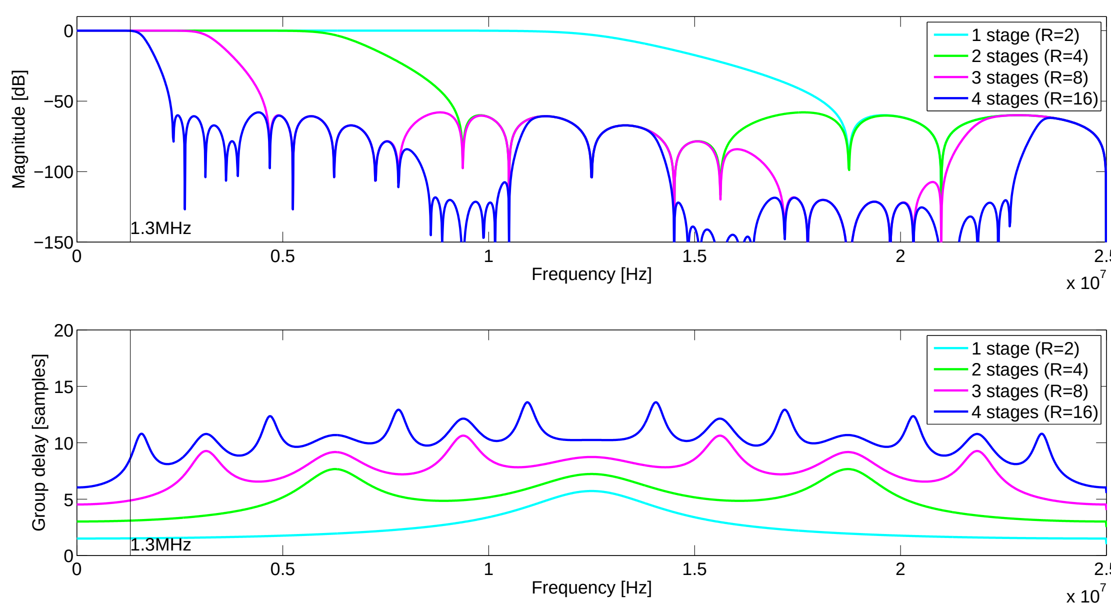


Figure 3.21: Frequency response and group delay of cascaded IIR half-band filters

With the elliptic IIR design the filter order could be decreased to five and with the help of polyphase decomposition the number of coefficients were further reduced to two. So the cascade of four half-band filters uses only eight coefficients. The stop-band ripple was specified to reach a value of -60 dB. This threshold is slightly violated about 2 dB what shows that the specification is barely realizable with a filter order of 5.

Concerning the group delay shown in figure 3.21, it can be determined that the group delay in general is much lower than presented in section 3.4.1.2 where IIR filters were designed with more or less the same specifications. This difference in group delay between elliptic IIR filters from section 3.4.1.2 and elliptic half-band filters in this example is a factor of about eight. The four half-band filters in cascade generate eight times less group delay variation. For one, two or three half-band stages the group delay introduced in the band of interest is negligible. With four stages in cascade the group delay, which changes approximately 2.6 samples from lowest to highest passband frequency, may have an impact on the signal quality.

Remark 3 *This section has shown that using half-band filters offers some rewards compared to simple FIR or IIR filter implementations. They are especially applicable in decimation filters where decimation factors with an integer power of two are required.*
FIR half-band filters have only half the number of coefficients and therefore save chip area and reduce power consumption in a hardware design. IIR half-band filters can even further reduce the number of coefficients but they introduce non-linear phase. Other aspects that should be kept in mind when using IIR filters are: sensitivity to coefficient quantization where instability can occur, limit cycles due to rounding effects and sufficient accuracy of filter coefficients that leads to high bit-widths of registers.
Power consumption in half-band filters is reduced by a stepwise rate decimation and polyphase decomposition that allows the filter to run almost completely at the lower sampling rate.

3.4.4 Wave Digital Filters

Analog filters in electronic circuits are based on voltage and current parameters. Signal flow graphs of these circuits are not realizable in discrete-time systems because an exactly equivalent implementation in the digital domain is not possible. Only approximations of the analog reference systems with worse properties are implementable. This is the case for conventional design methods like the impulse invariance design or bilinear transformation where impulse response or the system transfer function are used for transformation. Analog circuits, which can be described with the help of flow graphs, are not directly mappable to the digital domain because the following conditions are not satisfied [22]:

- Directed loops in the flow graph must not be delay free
- Any delay must be an integer number

To circumvent the problem of non realizable discrete-time systems Alfred Fettweis introduced a method [8] where wave theory is applied instead of Kirchhoff's current and voltage laws to transform analog systems to discrete-time systems. These new types of discrete time filters are called Wave Digital Filters (WDF). Because they are based

on the same analog reference filters like IIR filters, frequency and phase response are approximately the same. Only the way how the transformation is performed and which parameters are used is different.
WDFs showed up to have following advantages compared to their conventional IIR counter part implementations [19]:

- inherent stability
- low sensitivity to coefficient variation in filter passband
- low accuracy of multiplier coefficients required

Drawbacks of Wave Digital Filters according to [19] are more complex structures, which require more arithmetical operations compared to conventional IIR filters. Furthermore they are less modular.

Wave Theory: In transmission line theory voltage waves are described by an incident wave $\underline{A}$ and a reflected wave $\underline{B}$. Relative to the transmission line length x the reflection coefficient $\underline{\Gamma}$, describing the ratio between $\underline{A}$ and $\underline{B}$, varies. For a one-port network with port resistance equal to the characteristic impedance Z_0 and load impedance $\underline{Z_L}$ the reflection coefficient at position x = 0 is defined according to equation 3.22. Figure 3.22 illustrates the transmission line as a two-port and the load as one-port network.

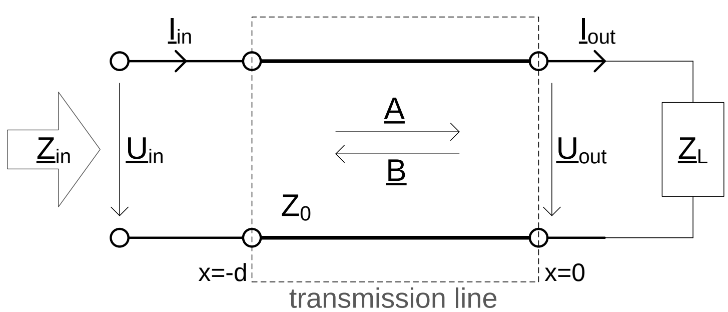


Figure 3.22: Transmission line characteristics

$$\underline{\Gamma}_0 = \frac{\underline{B}}{\underline{A}} = \frac{\underline{Z_L} - Z_0}{\underline{Z_L} + Z_0} \tag{3.22}$$

Transformation to Discrete-time Systems: The transformation to the digital domain is done via the Richards transformation [7] which is described by equation 3.23 where $\tau/2$ is the electrical propagation time on the transmission line.

$$\psi = \Sigma + j\Omega = tanh\Big(\frac{s\tau}{2}\Big) = \frac{e^{s\tau/2} - 1}{e^{s\tau/2} + 1} \tag{3.23}$$

It maps the s-domain into the so-called ψ-domain where the frequency ω is mapped $1/\tau$-periodic to the new domain space with frequency Ω [7]. This periodicity is shown in equation 3.24 where $s = j\omega$ was inserted into equation 3.23.

$$\psi = \Sigma + j\Omega = tanh\Big(\frac{j\omega\tau}{2}\Big) = j \cdot tan\Big(\frac{\omega\tau}{2}\Big)$$

$$\Omega = tan\Big(\frac{\omega\tau}{2}\Big) \tag{3.24}$$

A chain matrix can be postulated for a lossless commensurate-length transmission line like the one depicted in figure 3.22. It describes the correlation between input and output variables. After substitution with the Richards variable ψ, this chain matrix is defined according to equation 3.25 with $\underline{U}$ and $\underline{I}$ describing the complex input and output voltages and currents.

$$\begin{pmatrix} \underline{U}_{in} \\ \underline{I}_{in} \end{pmatrix} = \frac{1}{\sqrt{1-\psi^2}} \cdot \begin{pmatrix} 1 & Z_0\psi \\ \frac{1}{Z_0\psi} & 1 \end{pmatrix} \cdot \begin{pmatrix} \underline{U}_{out} \\ \underline{I}_{out} \end{pmatrix} \tag{3.25}$$

Figure 3.22 depicts the transmission line in combination with a load impedance at its output. The input impedance $\underline{Z_{in}}(\psi)$ of the whole network is defined by equation 3.26 and depends on the transmission line's output load $\underline{Z_L}$.

$$\underline{Z_{in}}(\psi) = \frac{\underline{U_{in}}}{\underline{I_{in}}} = \frac{\underline{Z_L} + Z_0\psi}{Z_0 + \underline{Z_L}\psi} \cdot Z_0 \tag{3.26}$$

There are three special conditions of $\underline{Z_L}$: matched terminated, open-ended and short-circuited. In the first case $\underline{Z_L}$ equals the characteristic impedance Z_0 of the transmission line so that wave reflections are prevented. The open-ended case has a resistance value of ∞ and reflects the wave with the same phase. Whereas the short-circuited case has zero resistance and reflects the wave with 180° shifted phase.

Next the input impedances of these three cases are inserted in equation 3.22 to express the reflection coefficient $\underline{\Gamma}(\psi)$ in the ψ-domain. The subsequent Z-domain transformation of $\underline{\Gamma}(\psi)$ is performed by equation 3.27 where z is defined by equation 3.28.

$$\psi = \frac{z-1}{z+1} \tag{3.27}$$

$$z = e^{s\tau} \tag{3.28}$$

Table 3.2 shows the resulting components in the ψ-domain, the equivalent input impedances and the reflection coefficients. The last column presents the Z-transformed elements.

ψ-component	$\underline{Z_{in}}(\psi)$	$\underline{\Gamma}(\psi)$	$\underline{\Gamma}(z)$
Resistance	Z_0	0	0
Capacitor	$\frac{Z_0}{\psi}$	$\frac{1-\psi}{1+\psi}$	z^{-1}
Inductor	$Z_0\psi$	$\frac{\psi-1}{\psi+1}$	$-z^{-1}$
Short-circuit	0	-1	-1
Open-ended	∞	1	1

Table 3.2: Relationship between ψ and Z-domain

Sections: The components presented in table 3.2 are used to build up Wave Digital Filters. Due to different port resistances these components are not directly connectable with each other. Therefore adapters which match the port resistance have to be used. They compose of adders, negators and multipliers with coefficients and connect two- or three-port components with each other. A combination of adapter and component is called a section. They can also be cascaded to obtain higher order sections. Observing the transfer function of a section it can be seen that they have all-pass behavior. The magnitude of the frequency response is one for all frequencies, only the phase is varying. A detailed discussion of WDF adapters can be looked up in [8] and [23].

Ladder structures, built of cascaded inductances and capacitances, tend to filter structures with three-port adapters whereas lattice structures need two-port adapters only. This advantage and the fact that they can be implemented more efficiently in hardware are reasons why Lattice Wave Digital Filters (LWDF) are more frequently used in practice than ladder filters [23]. Therefore only Lattice WDFs are discussed in this thesis.

3.4.4.1 Lattice Wave Digital Filter

Due to the complex theory of wave digital filters their design is not trivial. Fortunately some literature and a few tools that support the design process exist. A paper with formulas for calculation of LWDF coefficients has been presented by Lajos Gazsi in 1985 [9]. It shows a step-by-step method for calculation, derives the necessary formulas from wave theory and describes their relations.
A Matlab toolbox for coefficient calculation of LWDF filters has been published at University of Technology Delft [3]. It offers algorithms that support the user when designing wave digital filters. Additionally the toolbox can draw structural circuit plans of WDFs, calculate coefficients and plot frequency responses.

Lattice WDFs consist of two branches, each containing multiple sections in serial or parallel. It contains first and second order sections only. Each branch has all-pass characteristics and the desired frequency response is achieved by combining both branches at the filter output. This is done by simply adding their signal paths. Finally, at the filter output, the magnitude is halved.
Figure 3.23 shows an example of a 5^{th} order LWDF structure created with the *LWDF toolbox for Matlab* [3]. It consists of five sections each containing one coefficient γ. Low- or high-pass LWDFs must have odd filter orders. Delay units (z^{-1}) are labeled T in this figure. The diagram shows two separate filter outputs, one for high-pass and the other for

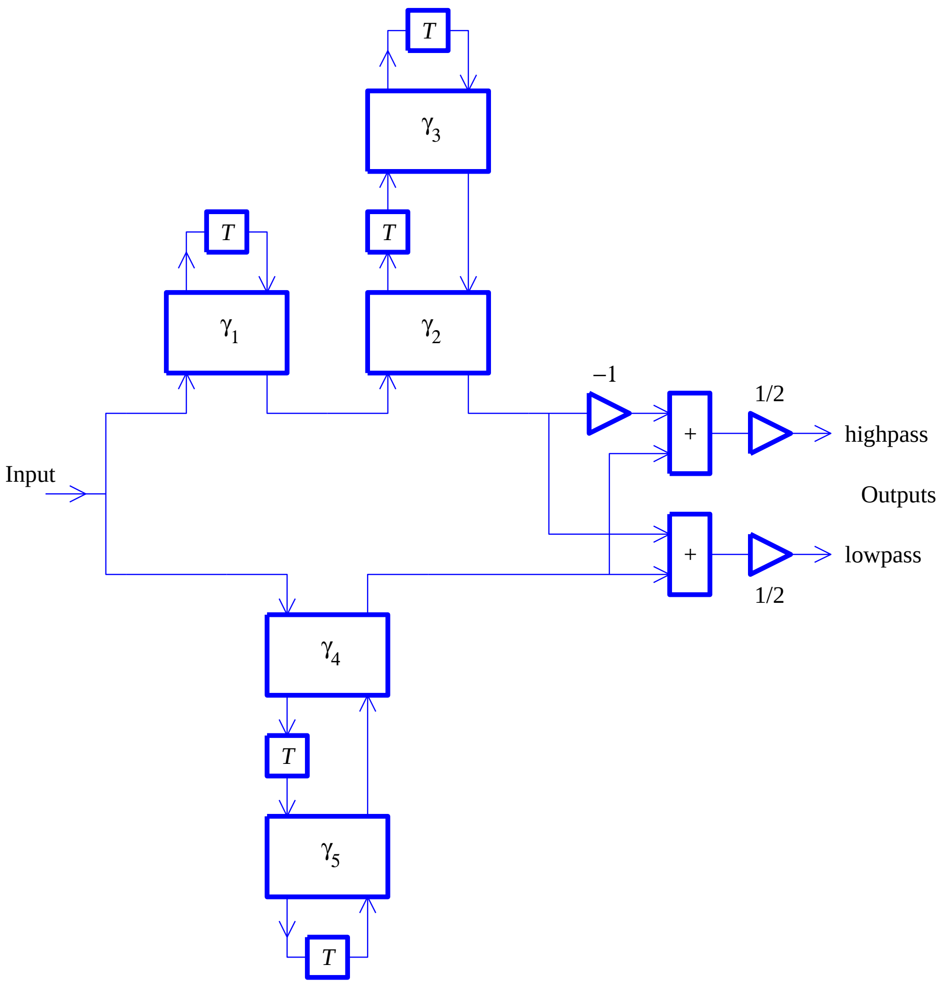


Figure 3.23: 5^{th} order Lattice Wave Digital Filter structure

low-pass implementations. An important point concerning these outputs is that only one matches the filter specifications, the other one does not have symmetrical characteristics. This is only the case for Bi-Reciprocal LWDF structures, which are described in section 3.4.4.2. The frequency response of both outputs is depicted in figure 3.24. This WDF was specified to be an elliptic filter with order five and stopband ripple of -60 dB.

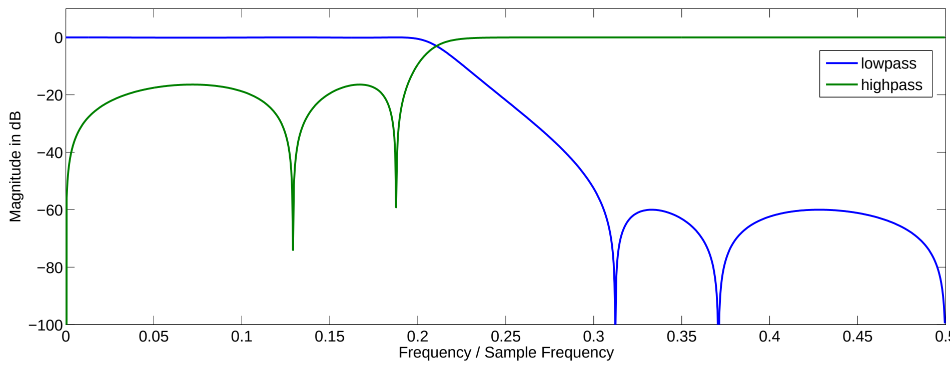


Figure 3.24: Frequency response of a 5^{th} order LWDF

3.4.4.2 Bi-Reciprocal Lattice Wave Digital Filter

This special type of LWDF belongs to the group of IIR half-band filters which were described in section 3.4.3. They have their cut-off frequency at half the Nyquist frequency and therefore are well suited for sampling rate decimation applications.

Again the *LWDF toolbox for Matlab* [3] has been used to design a 5^{th} order Bi-Reciprocal LWDF filter with -60 dB stopband ripple based on an elliptic reference filter. Due to the half-band characteristics each second coefficient is zero, hence all second order sections change to first order sections. The resulting structure is illustrated in figure 3.25 and is composed of two first order sections each containing one coefficient. Two delay elements are connected to each adapter.

The frequency responses of both filter outputs are shown in figure 3.26. It can be seen that the outputs have symmetrical frequency response around $\omega = \pi/2$ this time. The magnitude at this frequency point is -3 dB. Both outputs, low-pass and high-pass, fulfill the stopband ripple specification.
As said before, Bi-Reciprocal LWDFs, are perfectly suitable for rate decimation applications with factor of two. The structure from figure 3.25 can be modified with some minor changes to obtain a more efficient implementation suitable for rate decimation as depicted in figure 3.27.

By doing rate decimation at the beginning of the signal path the delay units in the adapters are changed to process each sample instead of every second. The first delay unit in the upper branch still has to run on the high frequency and introduces a delay of the upper path compared to the lower one. This has the effect that samples which are

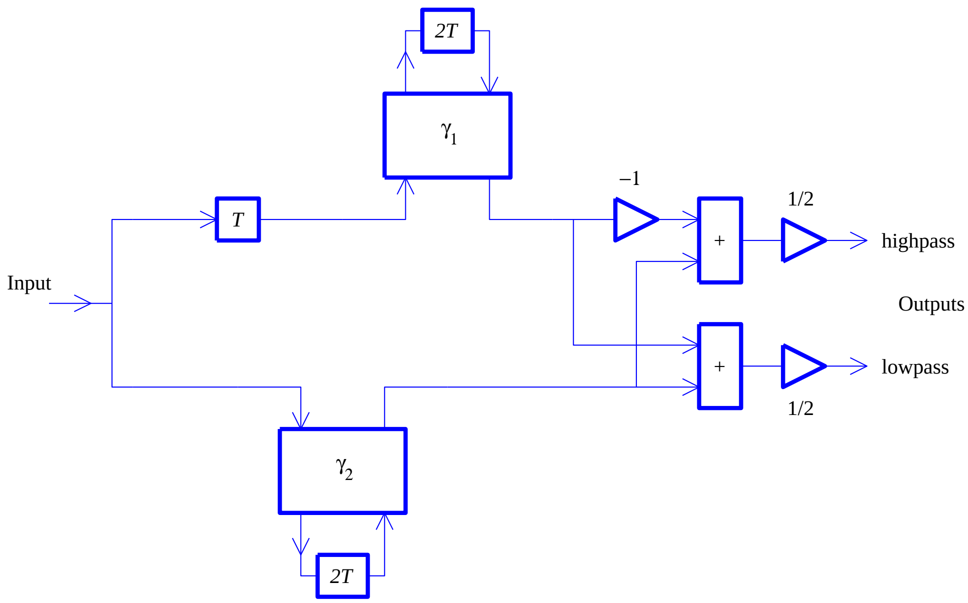


Figure 3.25: Signal path of a 5^{th} order Bi-Reciprocal LWDF

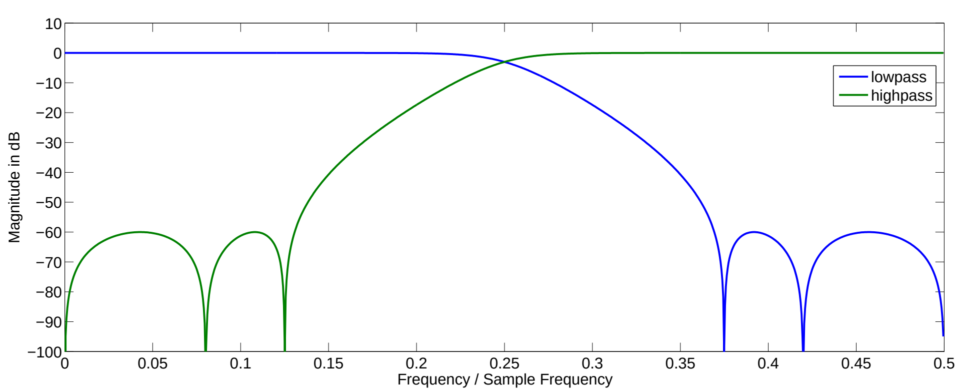


Figure 3.26: Frequency response of a 5^{th} order Bi-Reciprocal LWDF

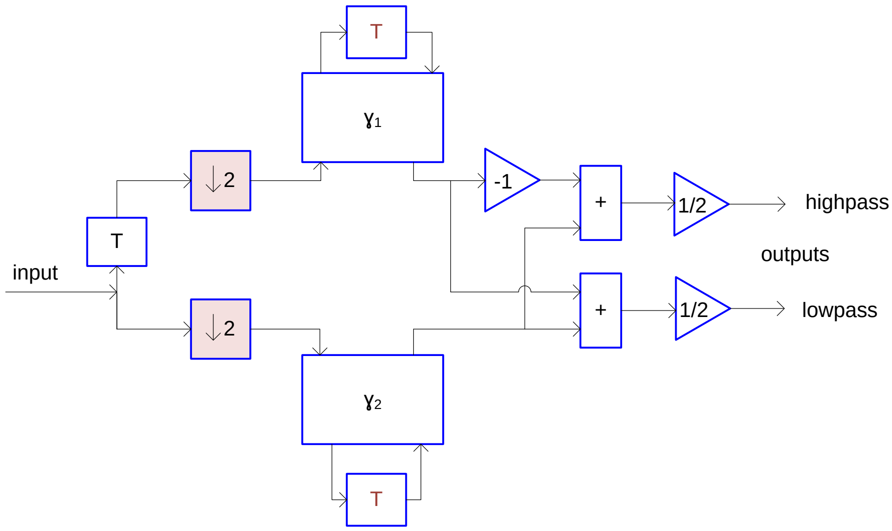


Figure 3.27: 5^{th} order Bi-Reciprocal LWDF with rate decimation factor of two

processed by the lower branch are thrown away in the upper one and vice versa. The major advantage of this modified structure is the early sampling rate decimation which of course helps reducing power consumption.

Remark 4 *In this section an alternative filter implementation with some remarkable properties like inherent stability and insensitivity to coefficient quantization has been presented. Although Wave Digital Filters do not come to mind when thinking about decimation filters, they are well suited for custom implementations in a digital integrated circuit design where power consumption and chip area matter. Due to their insensitivity to coefficient quantization and their low coefficient accuracy they can be implemented with lower bit-widths than conventional IIR filters. The Bi-Reciprocal LWDF is a IIR half-band filter that suits very well for rate decimation factors of two. With their improved properties, WDFs can compete conventional IIR half-band filters and lead to smaller hardware implementations with less power consumption.*

Chapter 4

Verification

4.1 Test Bench

For verification and evaluation purpose a test bench is implemented. Because simulations are done with Matlab and ModelSim, a combined concept is set up. Matlab is a convenient high-level scripting language with a huge variety of toolboxes and functions perfectly suitable for rapid signal generation as well as signal analysis. ModelSim is used for VHDL simulation with stimuli generated in Matlab. This combination makes the test approach easy to adopt and multi functional. Figure 4.1 shows a block diagram of the test bench.

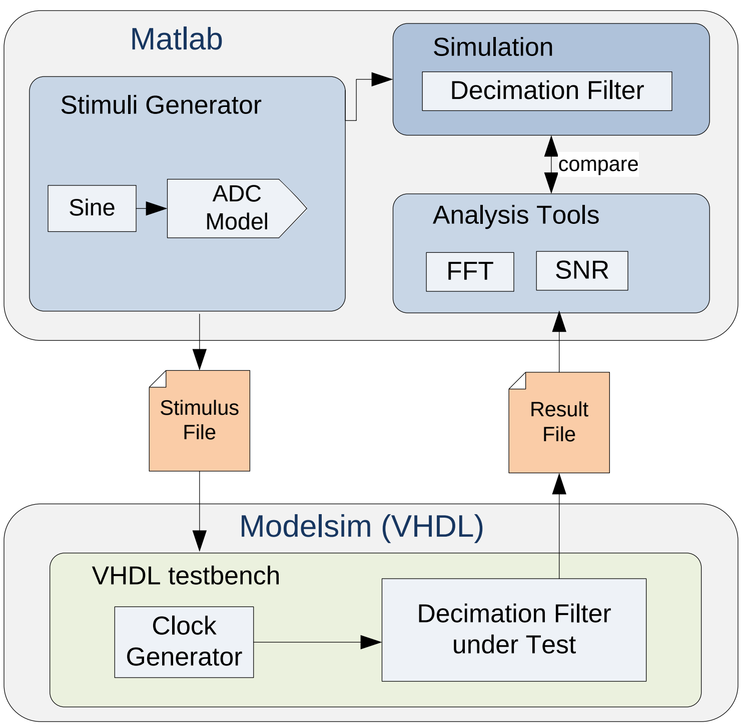


Figure 4.1: Test environment for verification and evaluation of VHDL modules

4.1.1 Test Procedure

First a high resolution test signal with maximum magnitude of ± 1 is generated in Matlab. It is used as input to the $\Sigma\Delta$ A/D converter model which delivers a 4 bits output signal with 50 Mega-samples per second. This is the stimulus for the VHDL implemented filter under test. The generated stimuli data is then written to a file for further processing within ModelSim. For accurate results it is essential that the filter is fed with the output signals of the $\Sigma\Delta$ A/D model instead of simple sine waves. The reason for this is the limited input bit-width that prohibits a straight forward sine wave stimulus because the signal quantization would be too high. For example a CIC filter with five bits input register could be stimulated with a five bits sine signal only.

The next step after stimulus generation is starting ModelSim for running the VHDL simulation of the decimation filter under test. Matlab passes the filename of the stimulus data as well as the simulation time to ModelSim. That runs a simulation of the VHDL test bench, containing the filter under test, with the stimulus data extracted from the file. Clock and control signals as well as filter parameters are defined in the VHDL test bench.

After successful simulation in ModelSim the output signal data is again written to a file. ModelSim is stopped and Matlab proceeds with the analysis of the results. This last step allows various analysis methods of the simulation results. One example is the SNR analysis which is described in section 4.2. In addition to the ModelSim results, filter implementations done in Matlab are fed with the same stimulus so that both results can be compared. This method is especially useful for verification of new VHDL designs.

4.1.2 Reference Channel Filter

It was already mentioned in section 3.3 that the implemented decimation filters are part of one long digital signal processing chain. Therefore it is sometimes not sufficient to interpret results at the output of the decimation filter only. For example when the quality of a decimation filter should be analyzed, but the frequency spectrum still contains a lot of noise or harmonics which influence the result. In the end, e.g. the signal quality in a channel is of interest and not the whole frequency band. Hence signal distortion and aliasing due to decimation should be observed within a channel. For this reason a nearly ideal band-pass filter, which simulates the behavior of the channel filter, is implemented in Matlab. It is applied to the measured data before analyzing is done. Instead of implementing two channel filters for both complex signal branches, as described in section 3.2, a simple real channel filter is used for simplicity.

Parameter	Value
filter type	FIR
phase	linear
stopband ripple	-60 dB
passband ripple	0.1 dB
bandwidth	50 kHz
transition bandwidth	50 kHz

Table 4.1: Parameters of the channel reference filter

For different test frequencies the filter's center frequency is adopted and the coefficients are recalculated. The bandwidth of 50 kHz as well as the transition bandwidth stays the same for all measurements. The output of the filter is quantized to 14 bits fixed point representation. Table 4.1 summarizes the filter parameters.

The frequency response of the channel filter for a sampling frequency of 5 MHz and a center frequency of 500 kHz is depicted in figure 4.2.

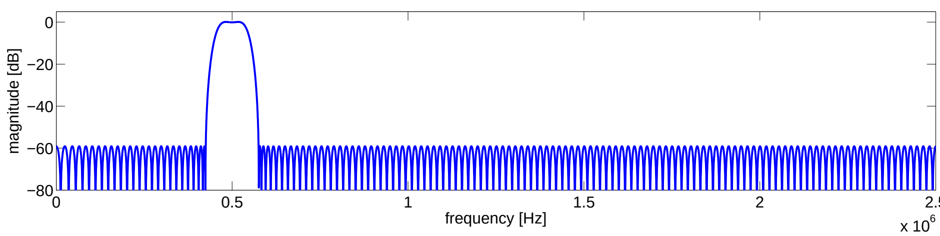


Figure 4.2: Frequency response of the channel filter

4.2 SNR Analysis

The Signal-to-Noise Ratio (SNR) is a quantitative measure for signal quality in many fields of applications. It defines the ratio between signal and noise power given in decibel. Related to this definition the dynamic range describes a range where information is contained within a signal. It is the ratio between the largest and the smallest possible value of a quantity in a signal.

In signal processing applications the SNR is often used to quantitatively measure and compare a signal's quality. In digital signal processing where signals are represented by fixed-point numbers the signal to quantization noise ratio of one bit is equal to 6.02 dB under the assumption of uniform noise distribution. If k bits are used, an SNR of $k \cdot 6.02\,dB$ is achievable.

In the following chapters SNR analyses of digital filters are performed in order to determine and compare their quality. Section 4.2.1 and 4.2.2 describe two different ways of SNR measurement. Both methods require a generated sine wave with constant frequency as test signal.

4.2.1 Minimum Sinusoidal Error Method

The Minimum Sinusoidal Error (MSE) method evaluates a signal's time-domain information and measures the deviation of the analyzed signal from the generated sine wave. The input signal frequency and the sampling frequency has to be known in order to do the analysis. For SNR calculations with the MSE method an algorithm from [23] can be used.

4.2.2 Spectral Analysis with Coherent Sampling

This method uses the Discrete Fourier Transformation (DFT) to analyze a signal's frequency spectrum where signal and noise components are classified. With the knowledge of the input signal's frequency a separation of signal and noise bins is possible. One problem of spectral analysis is caused by leakage. This inherent property of the DFT makes it difficult to measure a signal's magnitude because signal power leaks to frequency bins around. A method that prevents leakage in spectral analysis is called *Coherent Sampling* [2]. Thereby a periodic signal with frequency f_{sig} sampled at a sampling rate f_s must have an integer number M of periods which fit into a sampling window of N samples. This relation is described by equation 4.1.

$$N \cdot f_{sig} = M \cdot f_s \tag{4.1}$$

It was also mentioned in [2] that the number of samples should be mutually prime to the number of periods. This has the effect that the number of distinct sample phases within the period is increased and the results become more accurate.

Chapter 5

Implementation

5.1 Overview

In this chapter two different hardware implementations of decimation filters for employment in a linear receiver with $\Sigma\Delta$ A/D converter are presented. They were designed with respect to the specifications from section 3.3 and with conclusions obtained from chapter 3. The main focus was put on efficient filter structures for implementation in hardware to obtain small chip size and low power consumption. The written VHDL code follows the filter design process from this chapter in a register transfer level description. All modules have variable bit-widths for their in- and output ports, therefore they can be reused in other designs too. Some of the filters are even full parameterizable through generics. One example is the CIC module which uses *generate*-statements for building integrator and comb blocks dependent on the filter order.

The here presented filter implementations do not fulfill the requirements perfectly. This is not possible because of diverse properties and constraints that vary for different filter types. Beside some drawbacks and advantages both implementations are suited for the specifications made in section 3.3. In chapter 6 a detailed comparison between the implementations will be done, with respect to their advantages and disadvantages.

5.2 Tools and Design Flow

The implementation of the decimation filters was done in VHDL-87 due to the company's predefined design flow, but any Hardware Description Language (HDL) could have been used for this task. As development tool the text editor *Emacs* was chosen because of it's good integration in the design flow. Besides automatic generation of template code it offers an integration of *ClearCase*, a software configuration management tool with version control.
The project was set up with the Infineon specific design environment *InWay* [1] which includes the whole design flow as well as design packages. Simulations are done with *ModelSim* from *Mentor Graphics* and for synthesis *Design Compiler* from *Synopsys* is used.

5.3 Sharpened CIC Filter

When talking about Cascaded Integrator Comb filters in section 3.4.2 it was shown that they perform very well in decimation applications. Due to their simple and coefficient-less structure they are quite efficient. When trying to implement a CIC filter that matches the specification from section 3.3 it turns out to be not that easy. The main problem is the combination of stopband attenuation which requires high filter orders and passband droop that grows with the filter order. Figure 5.2(a) shows the passband droop without additional sharpening structures.
To circumvent this problem a compensation structure has been presented in section 3.4.2.4. With the help of this additional structure it is possible to design a CIC filter that satisfies the specifications. Moreover the filter has finite impulse response and does not introduce distortions due to its linear phase. Figure 5.1 shows the block diagram of the whole filter structure.

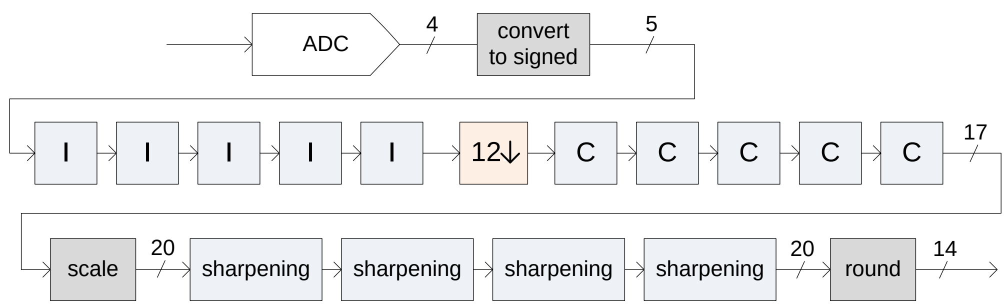


Figure 5.1: Sharpened CIC filter block diagram

5.3.1 Filter Design

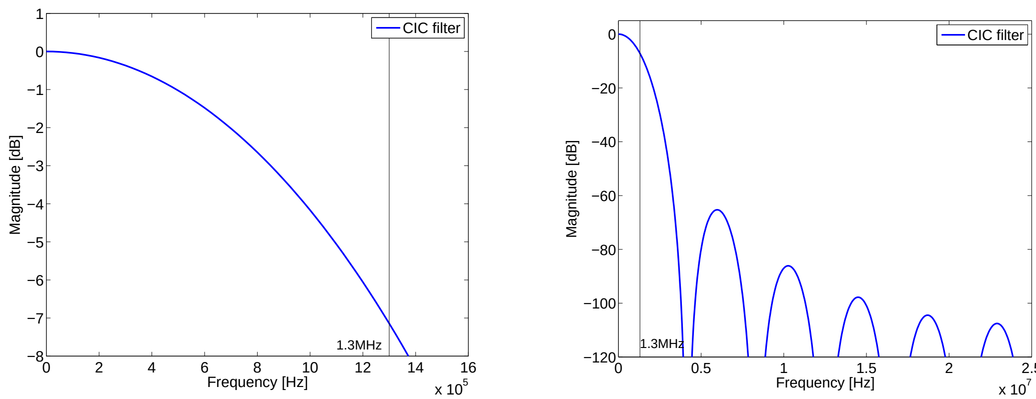
First of all suitable parameters for the CIC filter have to be found. They are chosen with respect to the specifications. In this step the passband droop of the filter is not regarded because the sharpening structure is applied afterwards to compensate the passband droop. Table 5.1 presents the parameters used for the CIC filter design.

differential delay	order	decimation factor
1	5	12

Table 5.1: Parameters of the CIC filter

The filter order is responsible for the high stopband attenuation and together with the decimation factor it influences the transition steepness and the passband droop. Higher filter orders would result in better stopband attenuation but also larger register bit-widths in the CIC filter as well as additional filter stages in the compensation filter. An order of five turns out to result in sufficient stopband attenuation.
With a decimation factor higher than 12 the passband attenuation after the compensation filter would be greater than 2 dB, so it was not further increased. By reducing the decimation factor to 10 the passband droop can be fully compensated. But here a higher

decimation is preferred in order to reduce power consumption.
The stopband ripple of the CIC filter is below -60 dB and the passband droop within the band of interest is about 2.32 dB at 750 kHz and 7.14 dB at 1.3 MHz (figure 5.2(b)). This very high passband droop, illustrated in figure5.2(a), will be compensated with the sharpening filter.



(a) CIC filter passband droop (b) CIC filter stopband attenuation

Figure 5.2: CIC filter frequency response

The CIC sharpening filter, introduced in section 3.4.2.4, is applied in cascade to the CIC filter output. Figure 5.3 shows how the frequency response is changed by the combination of both filters.

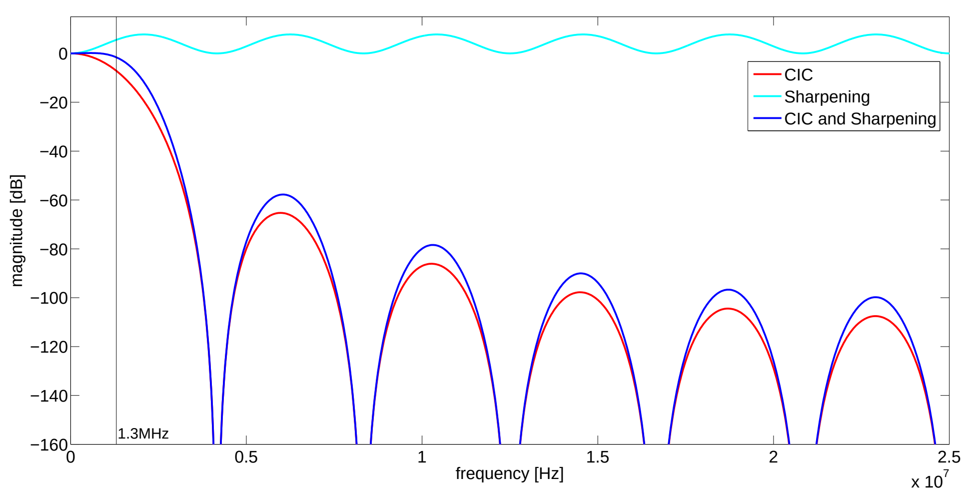
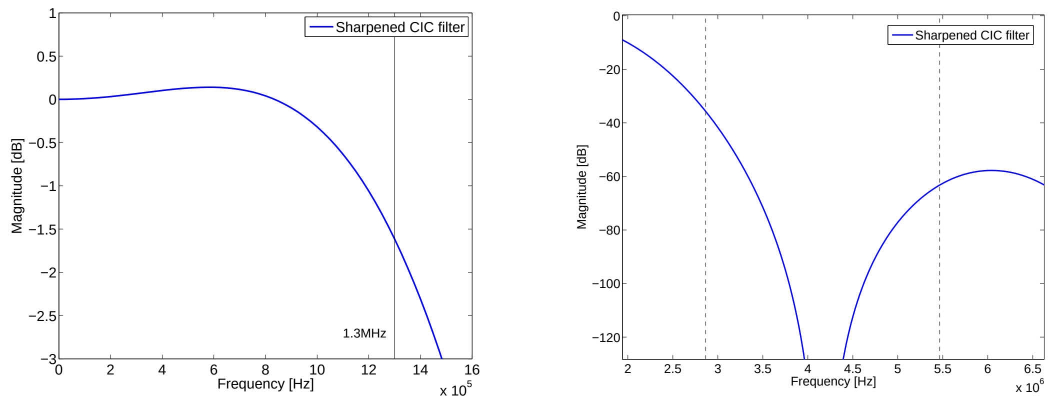


Figure 5.3: Frequency response of CIC and sharpening filter separately and in combination

It has filter order N-1, where N is the CIC filter's order. So 4 times the sharpening structure is applied to compensate the passband droop of the 5^{th} order CIC filter. One side effect of the sharpening filter is its gain at the CIC filter's side lobes exactly. So the stopband ripple of the CIC filter is raised about 8 dB which results in an increase of magnitude from -65.5 dB to -57.5 dB at the first side lobe. The filter's passband is sharpened to mitigate passband droop which is reduced to 1.6 dB at 1.3 MHz. Between 100 kHz and 800 kHz a slight filter gain up to 0.15 dB can be noticed. Figure 5.4(a) depicts this compensated passband. The frequency band that causes most aliasing in the band of interest is illustrated in figure 5.4(b). The folding band is indicated by the vertical dashed lines.



(a) Passband droop (b) Aliasing band

Figure 5.4: Sharpened CIC filter frequency response

The transfer function of the overall system consisting of equation 3.8 and 3.21 is combined in equation 5.1.

$$H(z) = H_{cic}(z) \cdot H_{sharp}(z)$$

$$H(z) = \Big(\sum_{n=0}^{11} z^{-n}\Big)^5 \cdot \Big(-\frac{1}{16}z^{-12} - 2z^{-24} - \frac{1}{16}z^{-36}\Big) \tag{5.1}$$

From the system transfer function it can be seen that this is a finite impulse response filter with linear phase. Concerning hardware implementation it is quite efficient because only a small number of coefficients are not equal to one. But even those are easy to implement with a few shift and add operations only.

5.3.2 Bit-Widths

Due to the ΣΔ A/D converter's resolution of 14 bits at a bandwidth of 10 kHz, the subsequent filter chain has to process signals with at least 14 bits to preserve signal information. This leads to an output resolution of 14 bit represented as signed integer. The input of the CIC filter is directly connected to the A/D converter which delivers four bits in unsigned representation at its output. This signal is scaled to five bits signed to avoid DC components. Equation 5.2 describes the scaling with sig_{adc} being the four bits ADC output and sig_{cic} the scaled CIC input signal.

$$sig_{cic} = (sig_{adc} \cdot 2) - 15 \tag{5.2}$$

With the help of input and output bit-width the register size of the CIC filter can be calculated. By inserting the filter parameters into equation 3.10 the maximum register growth is determined to be 23 bits (for signed integer). As described in section 3.4.2.1 the bit pruning method is applied to this example. The resulting bit-widths for each integrator and comb stage are presented in table 5.2. Altogether 26 bits are saved with this method.

I	I	I	I	I	C	C	C	C	C
23	23	23	23	21	20	19	18	17	17

Table 5.2: Bit-widths for each CIC filter stage

The output of the CIC filter is a 17 bits signed integer signal containing only 14 bit of information. Theoretically three bits could be truncated at this point but as the following sharpening filter requires some additional bits in order to improve the resolution of its calculations they are not. The reason for this measure is the sharpening filter's input stage which is performing a right shift by 4 bits, or a division by 16. Therefore, if the signal is represented by less than 18 bits, signal information is lost. Hence to maintain the signal quality at least 20 bits are necessary. SNR simulations for different bit-widths have been performed in order to choose an appropriate bit-width for the sharpening stage. As a measure for signal quality the character Q, which represents the mean SNR value over N measurements of different magnitudes from $0\,dB_{FS}$ down to $-72\,dB_{FS}$, is introduced in equation 5.3.

$$Q = 1/N \sum_{n=1}^{N} SNR_n \tag{5.3}$$

This method ensures that the whole dynamic range of the filter from the most significant bit down to the least significant bit is considered in the SNR analysis. According to section 3.3 the LSB is positioned at $-84\,dB_{FS}$ but for this measure the last two bits were neglected because they are assumed to contain the most noise components.

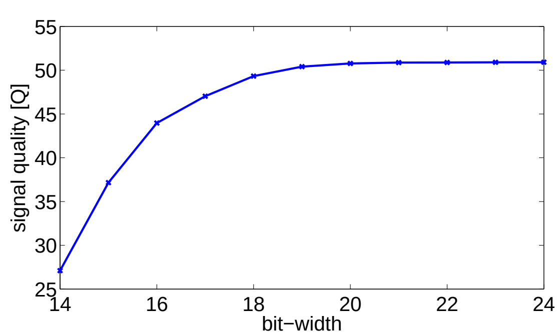


Figure 5.5: SNR simulation for various bit-widths of the Sharpening stage

Results of these measurements are illustrated in figure 5.5. The need for at least 18 bits in order to maintain the signal can be concluded from the bad signal quality resulting from 14 to 18 bits. Using more than 20 bits does not pay off any more. So the sharpening filter is working with bit-widths of up to 20 bits. The actual bit-width of each component within one sharpening stage is depicted in figure 5.6. Altogether one stage needs only 26 flip-flops, 60 full-adders and one inverter. The coefficients are implemented as simple shift operations.

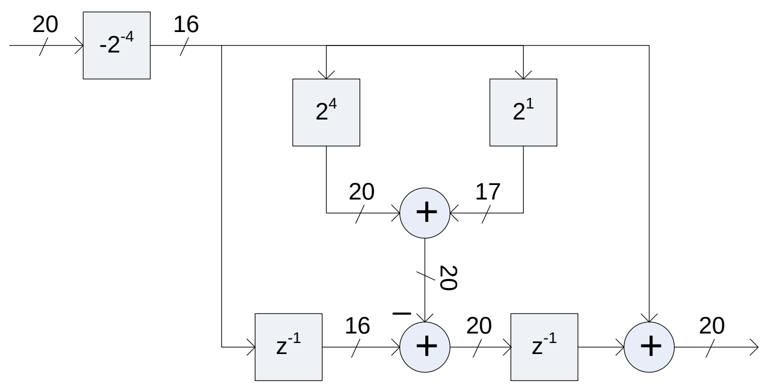


Figure 5.6: Sharpening stage bit-widths

5.3.3 Hardware Effort

The overall size of the digital design depends on several factors, e.g. process technology, number of coefficients and register bit-widths. Often absolute values of a digital design's area requirements are not really significant because they are connected to a certain process technology and size. Therefore it is common to describe the hardware effort in terms of building blocks that are used to achieve a certain task. For example, flip-flops that are used for storage and combinatorial logic that is performing calculations. This makes a comparison of the hardware effort possible.

In table 5.3 the effort for building the Sharpened CIC filter is summarized. The number of flip-flops and full-adders used in each module within the filter are itemized. Flip-flops are written first, the number of adders are specified in braces. Subtractions performed by adders are specified as full-adders in the table and only components considered in the design are counted. Because decimation filters have components running on different frequencies, columns separate these frequency domains to show which components are clocked higher and hence consume more power than others. This filter has two different clock rates f_s: 50 MHz and 4.167 MHz.

When looking at the sum of components implemented at lower clock rate there are more than twice as many flip-flops and adders compared to the higher rate. This is favorable for a low power design. What is further concluded from table 5.3 is that the compensation filter requires more adders than the whole CIC filter.

Module	f_s	$f_s/12$
CIC		
Integrators	113(113)	-(-)
Combs	-(-)	91(91)
Sharpening		
4 Structures	-(-)	144(240)
Sum	113(113)	235(331)

Table 5.3: Usage of flip-flops(full-adders) in relation to clock rates

5.3.4 Filter Verification

The performance of the implemented filter is verified by SNR simulations described in section 4.2 using the test bench introduced in section 4.1. For evaluation of the whole dynamic range the SNR analysis is performed for input magnitudes from $0\,dB_{FS}$ down to -84,dB_{FS}. Three simulations with different frequencies are generated. Figure 5.7 shows the results over the whole frequency band for three sine waves with different input frequencies. It is noticeable that the three curves are not straight lines as one would expect

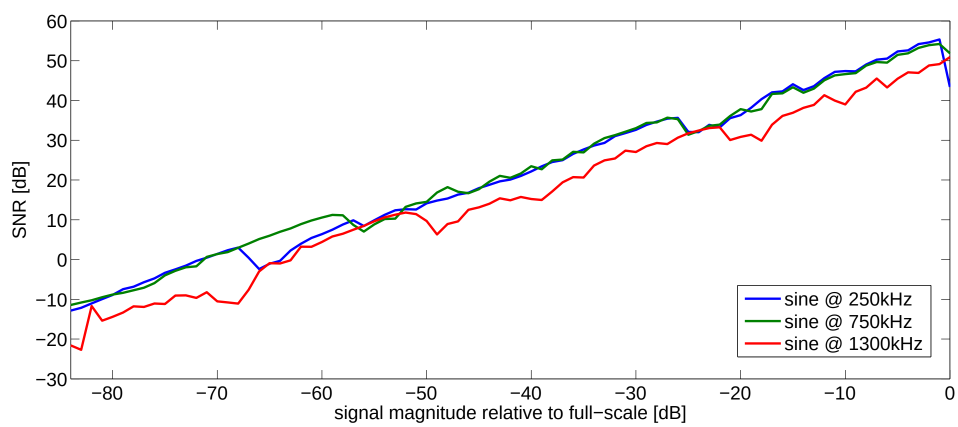


Figure 5.7: SNR in relation to signal magnitude for three input signals with different frequencies

at first. The reason for this is the remaining noise combined with harmonics and signal inter-modulations that still exist in the frequency band. One thing that has to be mentioned at this point is the SNR drop at $0\,dB_{FS}$ which can be explained by the non-linear behavior of the A/D converter with full magnitude.

For 250 kHz and 750 kHz the SNR curve is quite similar which shows that frequencies within the channel band are processed with approximately the same SNR. The curve at 1 300 kHz shows a loss of SNR of a few decibel. This can be traced back to passband droop, aliasing and ADC noise that are mainly affecting the upper frequencies in the band of interest.

In figure 5.8 the simulation is performed with an additional channel filter at the output of the test bench as described in section 4.1.2. The curves reflect the SNR within an equivalent channel and show a more flattened result because noise and harmonics outside the 50 kHz band-pass filter are attenuated. For these simulations three frequencies within the channel band from 250 kHz to 750 kHz are used because the band below 250 kHz and above 750 kHz is not intended for communication channels.

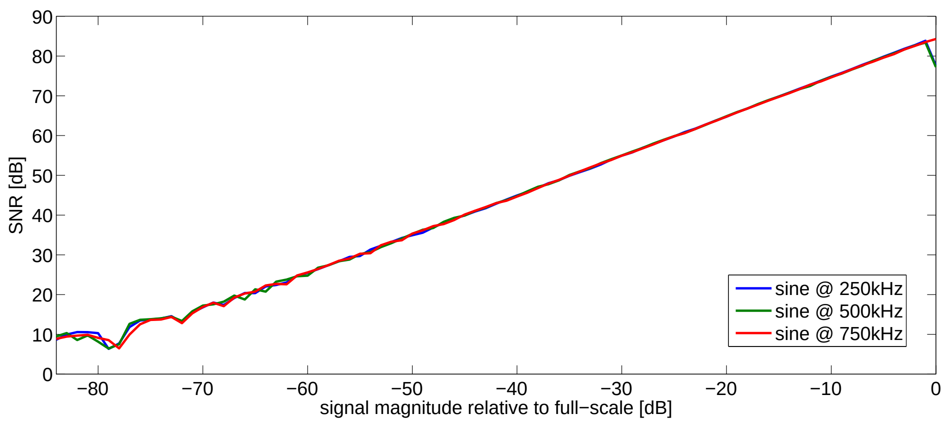


Figure 5.8: SNR simulation of three input signals with different frequencies using a channel filter

5.4 CIC and Bi-Reciprocal LWDF

This section describes a multi stage decimation approach implemented as combination of CIC and Bi-Reciprocal wave digital filter. The resulting filter has a sharp cut-off frequency, is inherently stable and robust against coefficient quantization. Drawbacks of this filter are the non-linear group delay and the passband droop that is introduced in the band of interest. Due to the non-linear group delay the maximum decimation factor is limited. Through higher decimation, distortion would be introduced in the band of interest.

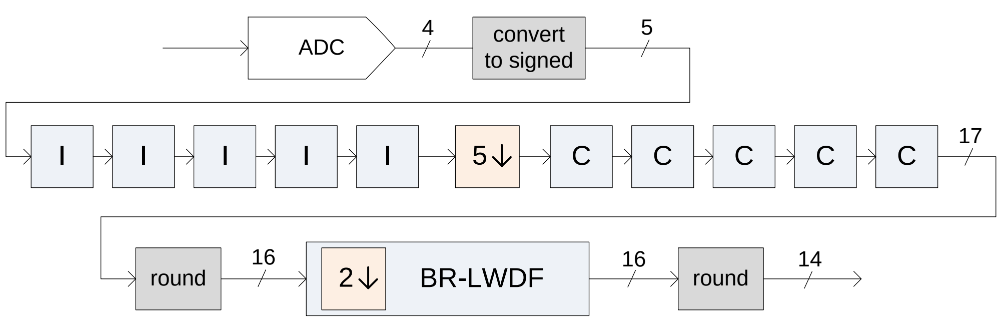


Figure 5.9: CIC and BR-LWDF filter block diagram.

A block diagram of the filter structure is depicted in figure 5.9. It shows the recursive CIC filter with decimation factor of five and its integrator and comb sections. Again

a recursive implementation was preferred because of the decimation factor that is not suitable for non-recursive implementations. It is not an integer power of two.
The Bi-Reciprocal LWDF does rate decimation by a factor of two and follows the CIC filter in the signal chain. Its non-linear group delay introduces distortions in the frequency band around the cut-off frequency.

5.4.1 Filter Design

The parameterization of the CIC filter, shown in table 5.4, is mainly influenced by keeping the passband droop low. For a 5^{th} order implementation, which attenuates the stopband by 60 dB, a rate decimation larger than five would result in high passband droop. With a decimation factor of five the attenuation is about 0.8 dB.

differential delay	order	decimation factor
1	5	5

Table 5.4: CIC filter parameters

With a factor of 5 in the CIC filter and 2 in the Bi-Reciprocal LWDF an overall decimation factor of 10 is achieved. The corresponding sampling rate is 5 MHz and only slightly higher than the sampling frequency of the Sharpened CIC implementation. In figure 5.10 the combination of both filters in terms of frequency response is illustrated. The periodicity of the Bi-Reciprocal LWDF shows up due to the lower sampling frequency after decimation performed by the CIC filter. Because of it's non-linear phase the Bi-Reciprocal LWDF has non-linear group delay which is illustrated in figure 5.11. The overall group delay is the sum of the CIC filter's constant and the Bi-Reciprocal LWDF's non-linear group delay.

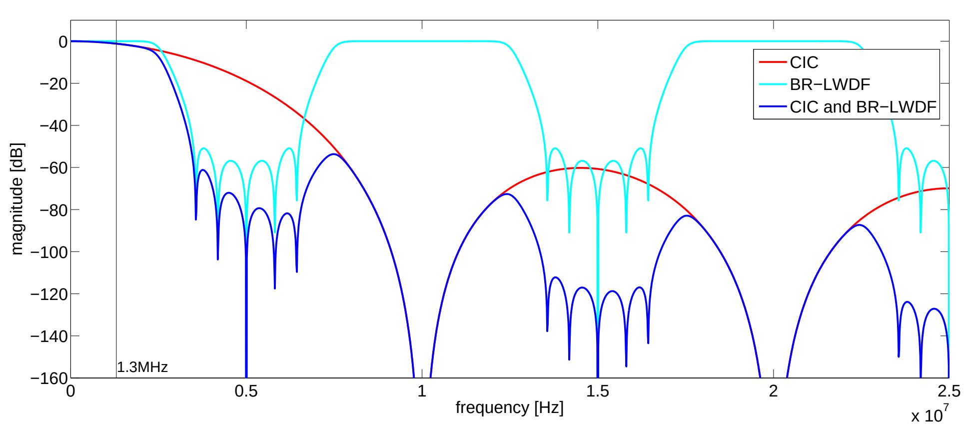


Figure 5.10: Frequency response of each filter separately and combined

The Bi-Reciprocal LWDF was designed to reach a stopband attenuation around -60 dB with two coefficients only. Due to the fact that Bi-Reciprocal LWDFs are half-band filters, half of their coefficients are zero and can be neglected in the filter design. With an filter order of five the desired result is achieved.

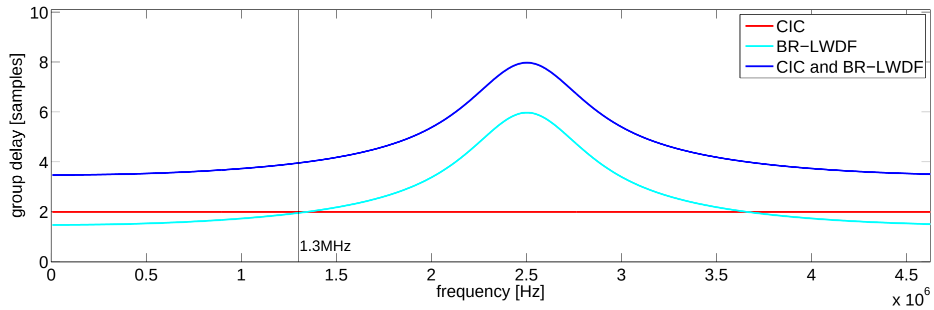


Figure 5.11: Group delay of each filter separately and combined

A structure for implementation of Bi-Reciprocal LWDF has been introduced in chapter 3.4.4.2. It requires odd filter order and consists of second order sections. Because each second coefficient equals zero, one of the two adapters in a second order section is canceled out. What remains are two delay elements and one adapter. One of the delay elements can be removed by placing the decimation factor in front of the adapters. Summarized, for a 5^{th} order filter implementation two adapters are needed, each containing one coefficient γ. With the help of the *LWDF Toolbox for Matlab* [3], filter coefficients for the Bi-Reciprocal LWDF are calculated. The implementation of coefficients in hardware is described in section 5.4.3 in more detail.

Figure 5.12(a) shows the passband droop of the overall filter implementation. The Bi-Reciprocal LWDF does not influence the passband significantly, hence only the CIC filter introduces passband droop. When having a closer look at the overall frequency response in figure 5.10, the highest stopband ripple can be determined to be around -53.5 dB in magnitude. This value is 6.5 dB higher than specified and will introduce higher aliasing. But fortunately the peak of the ripple is located 2.5 MHz away from the aliasing band that will fold down into the band of interest. This situation is illustrated in figure 5.12(b) which shows the aliasing band that introduces the most noise in the band of interest. The dashed lines indicate the aliasing band.

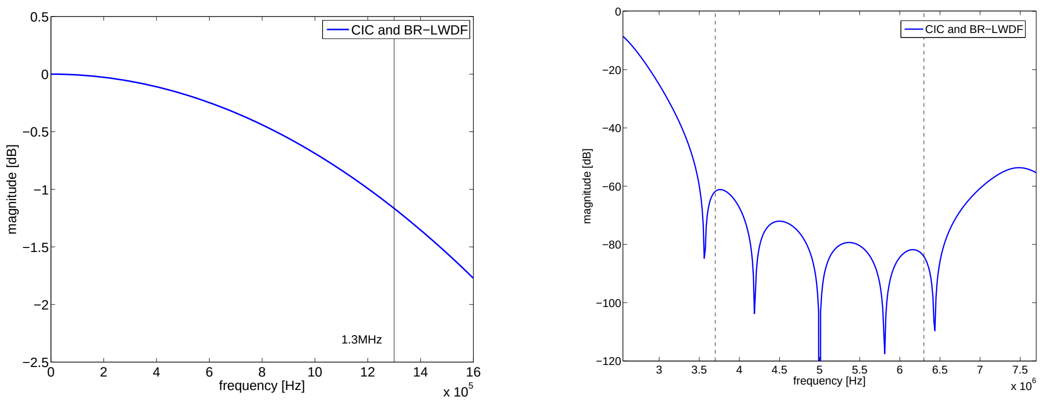
The transfer function of the overall system is shown in equation 5.4. It consists of the CIC filter and the combination of upper and lower branch of the BR-LWDF.

$$H(z) = H_{cic}(z) \cdot H_{br-lwdf}(z)$$

$$H(z) = \Big(\sum_{n=0}^{4} z^{-n}\Big)^5 \cdot \Big(z^{-5}\,\frac{-\gamma_1 + z^{-10}}{1-\gamma_1\, z^{-10}} + \cdot \frac{-\gamma_2 + z^{-10}}{1-\gamma_2\, z^{-10}}\Big) \tag{5.4}$$

5.4.2 Bit-Widths

Again the output signal of the A/D converter has to be scaled and shifted as described before in equation 5.2. The resulting input signal to the CIC filter is a five bits signed integer without DC components.



(a) Passband droop

(b) Aliasing band

Figure 5.12: CIC and BR-LWDF combined frequency response

CIC Filter: By applying input and output bit-width to equation 3.10 the maximum register growth of the CIC filter is calculated. It is determined to be 17 bits for a signed integer representation. According to section 3.4.2.1 further bit reduction is possible. But when the bit pruning equations are applied to this example it turns out that no more bit saving can be done at any stage. The relatively small input bit-width, which is also responsible for the low register growth, is the reason for this result. Hence each stage has to be implemented with 17 bits as shown in table 5.5.

I	I	I	I	I	C	C	C	C	C
17	17	17	17	17	17	17	17	17	17

Table 5.5: Bit-widths for each CIC filter stage

The maximum magnitude gain of the CIC filter is determined by equation 3.9 and results in 3 125. Combined with the maximum input magnitude of $2^4 - 1$, the maximum signal magnitude is calculated to be 46 875. This number in signed representation requires 17 bits. When reducing the bit-width to 16 bits by shifting the register by 1 bit to the right, the maximum magnitude is 23 438. Although this number needs 16 bits for representation, the highest bit is not fully exploited as shown in equation 5.5.

$$log_2(23\,438) = 14.5166 \tag{5.5}$$

In order to efficiently use the whole range of the register the signal needs to be scaled by a factor of 1.5 to reach a usage of 14.9 bits. Fortunately this factor is easy to implement in hardware with one adder and one shift operation. Equation 5.6 shows the structure where x is the signal that should be scaled and >> describes the bit-shift operation.

$$x \cdot 1.5 = x + (x >> 1) \tag{5.6}$$

Bi-Reciprocal LWDF: For the internal structure of the Bi-Reciprocal LWDF a bit-with of 16 has been determined by SNR simulations. Therefore the output of the CIC filter is quantized from 17 to 16 bits and fed into the LWDF. The bit-width of the signal path as well as the implementation of the adapters are shown in figure 5.13. Bold arrows indicate the signal path of the adapters. The upper adapter is suitable for γ-coefficients in the range from -1 to -0.5 and the lower one for γ-coefficients between -0.5 and 0. For simplicity the coefficient multiplications are drawn as simple multiplication. In hardware the Horner scheme, described in [23], is used to implement an efficient multiplication with simple structures. Section 5.4.3 depicts these hardware structures.

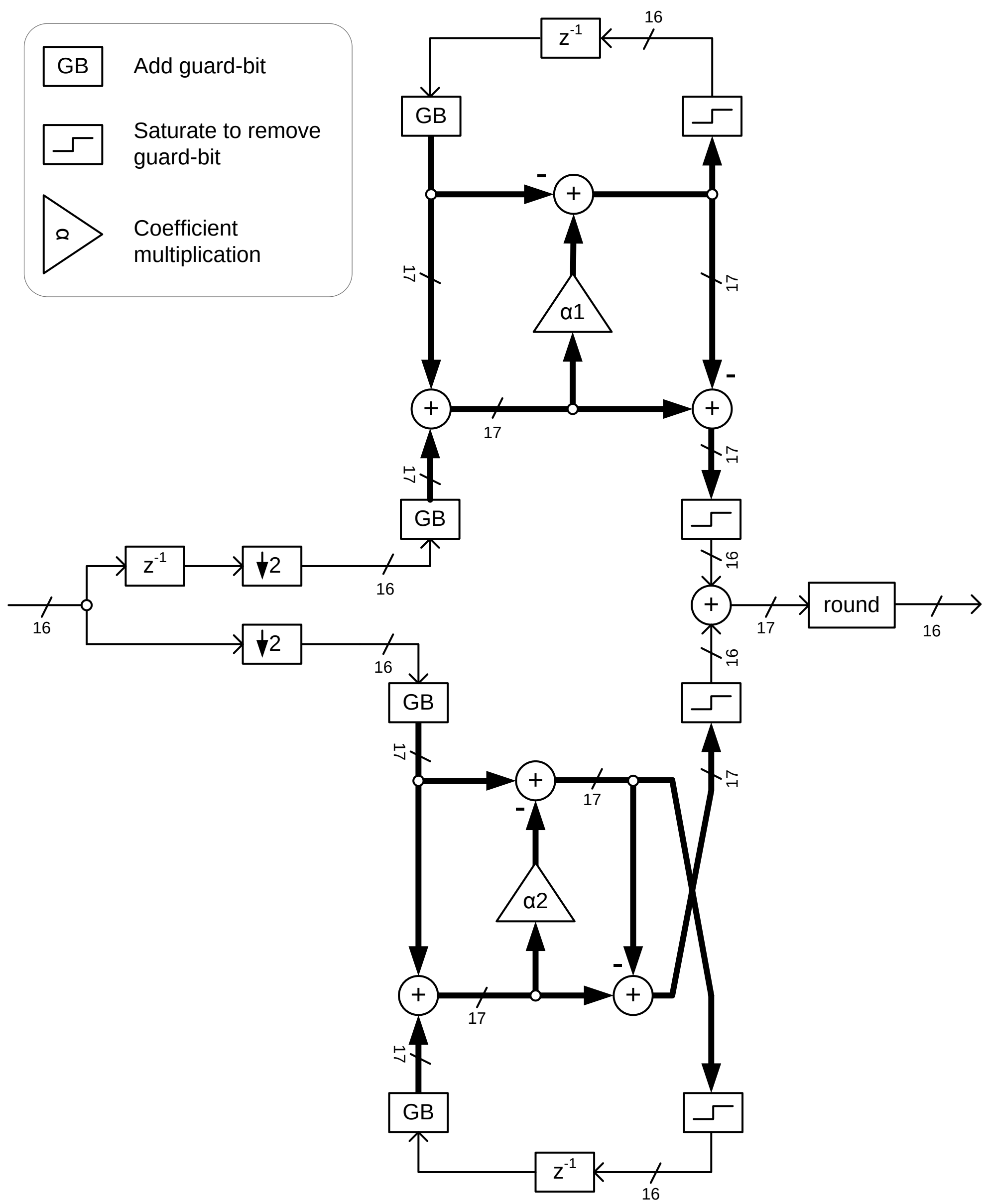


Figure 5.13: Bi-Reciprocal LWDF hardware implementation

5.4.3 Coefficients

For coefficient calculation the *LWDF Toolbox for Matlab* [3] was used. The result is presented in table 5.6 as γ- and α-coefficients. γ-coefficients are transformed to α-coefficients which can be implemented in a more efficient adapter structure according to [23]. Higher dynamic range of the signal due to better scaling is one of the advantages mentioned. Another reward is a reduction of hardware effort. The transformation to α-coefficients is done by the rules described by equation 5.7.

$$\begin{aligned} \alpha = |\gamma| \quad for \quad -0.5 \le \gamma < 0 \quad or \quad 0 < \gamma \le 0.5 \\ \alpha = 1 - |\gamma| \quad for \quad -1 < \gamma < -0.5 \quad or \quad 0.5 < \gamma < 1 \end{aligned} \tag{5.7}$$

index	γ-coefficient	α-coefficient	shift-operations	adders
1	-0.60888671875	0.39111328125	4	3
2	-0.152587890625	0.152587890625	4	3

Table 5.6: Bi-Reciprocal LWDF coefficients

In each case the hardware implementation of the adapter is different. Figure 5.13 in section 5.4.2 shows both structures that were implemented in this filter design. The hardware implementation of the α-coefficients is done with Horner's algorithm. It replaces a multiplication with shift and add operations. This is shown in equation 5.8 where x is the signal that is multiplied with the coefficient.

$$\begin{aligned} x \cdot \alpha_1 = (x - ((x - ((x + (x\,/\,32))\,/\,8))\,/\,4))\,/2 \\ x \cdot \alpha_2 = (x + ((x - ((x - (x\,/\,16))\,/\,8))\,/\,4))\,/\,8 \end{aligned} \tag{5.8}$$

In figure 5.14 the hardware implementation of both coefficients is illustrated where << and >> define bit-shift operations. To replace one multiplication, three shift and four add operations are required. The Horner algorithm cannot be performed in parallel so the whole combinatorial logic is evaluated in one clock cycle. This could lead to critical paths so that additional registers may be inserted in order to eliminate setup violations.

5.4.4 Limit Cycles

The first order adapters used in the design of Bi-Reciprocal WDF do not cause zero-input limit cycles. For each range of the γ-coefficients there exists another adapter implementation with this property according to [23]. But when using the Horner algorithm for coefficient multiplication, limit cycles may occur. It is mentioned that dependent on the coefficient range additional mechanisms have to be applied to get rid of limit cycles. Furthermore the output of the Horner scheme with -1 at it's input has to be investigated. For negative coefficients an output of -1 is correct but for positive coefficients this would result in limit cycles. Fortunately the coefficient γ_1 of the Bi-Reciprocal WDF leads to a limit cycle free filter structure if the output of the Horner scheme is -1 for an input of -1. This is the case for the structure shown in figure 5.14(a). The other coefficient (γ_2) can

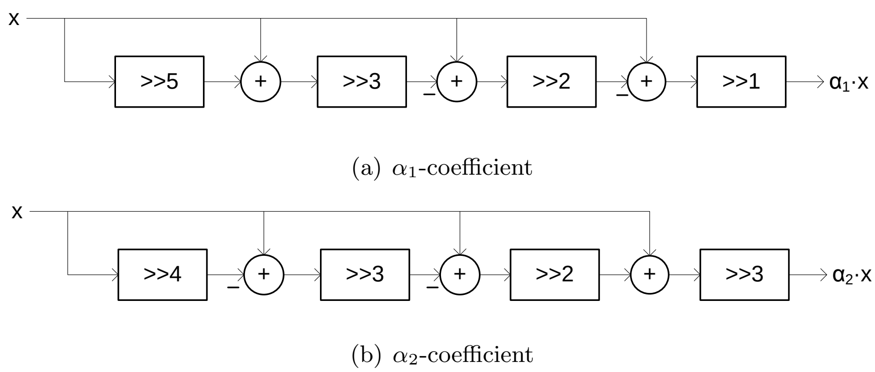


Figure 5.14: Hardware implementation of the coefficients

be implemented in an adapter structure that is inherently limit cycle free. Therefore no counter measures have to be taken as well.

For other filter coefficients and adapter structures certain mechanisms for getting rid of zero-input limit cycles have to be applied. With the following two methods first order adapters as they are used in Bi-Reciprocal LWDFs can be designed to be free of zero-input limit cycles [23]:

- Append Rear-bits
- Click-Clack

Rear-bits are additional bits that are appended at the least significant bit for higher resolution. It is added when the signal enters an adapter and removed when it leaves the adapter. The main purpose of the rear-bit is to eliminate values that prevent the signal from returning to zero after the input is zero. This can happen in the Horner scheme for certain constellations. Therefore the rear-bit is set to zero right after the coefficient multiplication with Horner's algorithm.
Another mechanism to prevent zero-input limit cycles is called Click-Clack. This method toggles the sign of the quantized signal before and after the quantization. Because quantization always results in the value that is nearer to $-\infty$, negative values will never reach zero. Thereby the signal is once quantized with and once without inverted sign. The result of both quantizations is different and allows the signal to reach zero.

5.4.5 Hardware Effort

The hardware effort of the CIC and Bi-Reciprocal LWDF is again estimated with the analysis of flip-flop and adder usage in the filter design. The same procedure has been done previously for the Sharpened CIC filter in section 5.3.3. In table 5.7 the number of used flip-flops is written first and the required full-adders are specified in braces. Three columns separate the three different clock rates the components are running at: 50 MHz, 10 MHz and 5 MHz.

It can be concluded from table 5.7 that the design has the most components in one of the two decimated frequency domains. This is essential for holding the power consumption at a low level. Furthermore the high number of adders in the last stage can be lead back to the coefficient implementation in the Horner scheme.

Module	f_s	$f_s/5$	$f_s/10$
CIC			
Integrators	85(85)	-(-)	-(-)
Combs	-(-)	85(85)	-(-)
BR-LWDF			
Upper branch	-(-)	16(-)	16(51)
Lower branch	-(-)	-(-)	16(51)
Coefficients	-(-)	-(-)	-(108)
Sum	85(85)	101(85)	32(210)

Table 5.7: Usage of flip-flops (full-adders) in relation to clock rates

5.4.6 Filter Verification

The performance of the implemented filter is again verified by SNR simulations which are described in section 4.2. Figure 5.15 shows SNR simulations of three input sine waves with different frequencies within the band of interest. It shows similar behavior to the simulations of the Sharpened CIC implementation. But due to the lower decimation factor less noise is filtered out and so the SNR curve is some decibel lower for the whole magnitude range.
The simulation of the 1 300 kHz sine wave shows again a deviation of some decibel compared to the frequency band from 250 kHz to 750 kHz.

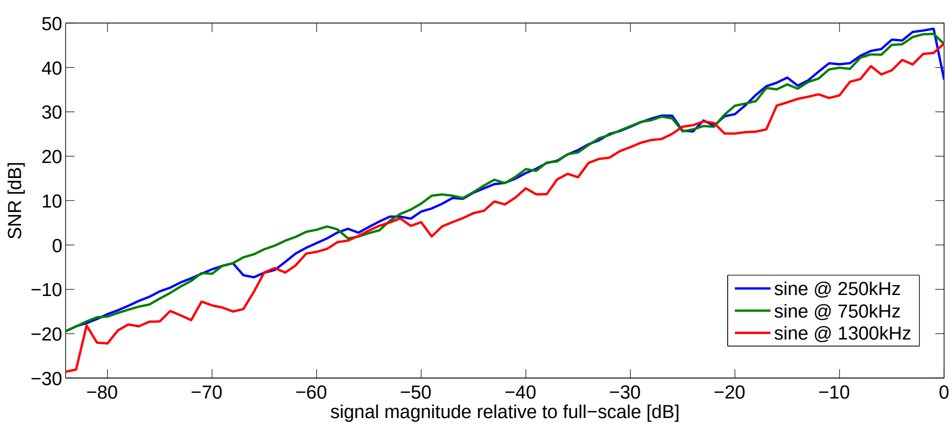


Figure 5.15: SNR in relation to signal magnitude for three input signals with different frequencies

The results with the reference channel filter are very similar to that of the Sharpened CIC filter. Due to band-pass filtering noise and harmonics around the channel are attenuated and the SNR values increase significantly. The simulation has been performed for frequencies within the channel band only.

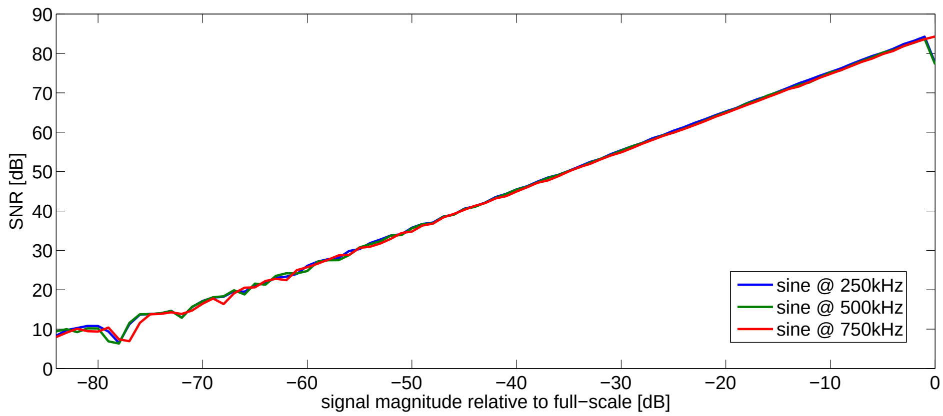


Figure 5.16: SNR simulation for three input signals with different frequencies using a channel filter

Chapter 6

Results

6.1 Comparison of Implementations

In chapter 5 two hardware implementations of decimation filters were discussed. Now their results are compared with each other in order to find one implementation for application in a linear receiver. Besides a comparison of the filter characteristics also SNR simulations as well as area and power estimations are performed.

Table 6.1 gives a rough overview of the different filter properties. One important fact that has to be kept in mind when comparing the filter implementations is the different decimation factor.

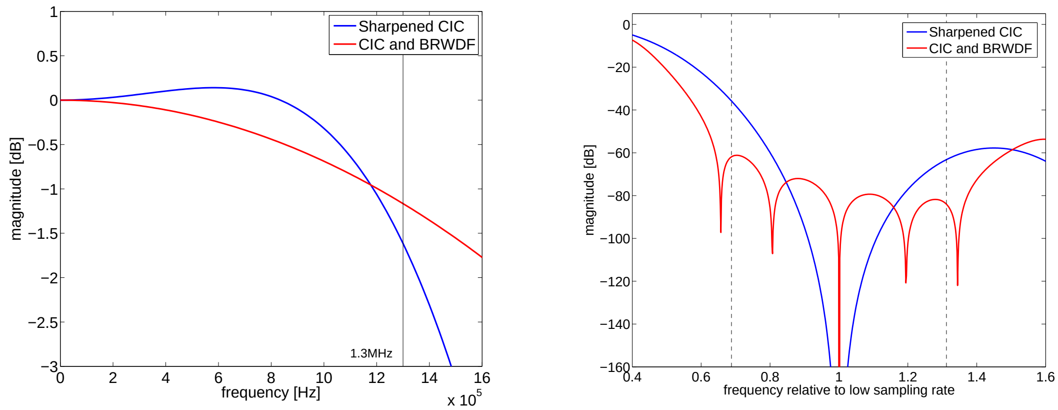
Filter implementation	Sharpened CIC	CIC and Bi-Reciprocal
Decimation rate	12	10
Worst aliasing attenuation	-35 dB	-61 dB
Phase	linear	non-linear
Lowest stopband attenuation	-57 dB	-53 dB
Hardware effort	higher	lower
Power consumption	higher	lower

Table 6.1: Comparison of filter implementations

6.2 Filter Characteristics

The frequency responses of both filter implementations are observed with regard to the passband (figure 6.1(a)) as well as the aliasing band (figure 6.1(b)) that folds down into the band of interest. Thereby figure 6.1(b) shows the the aliasing band that introduces most noise power in the wanted signal band. Two dashed lines to the left and to the right of the decimated sampling frequency indicate the aliasing band. Usually the left border indicates the larger noise amount.

Figure 6.2 shows the whole frequency response of the filters. The stopband attenuation is observed to be some dB higher for the CIC and Bi-Reciprocal LWDF. When comparing



(a) Passband (b) Aliasing band

Figure 6.1: Comparison of passband and aliasing bands

both filters' transition bands it can be seen that the Sharpened CIC filter starts earlier with the transition which results in a higher noise attenuation.

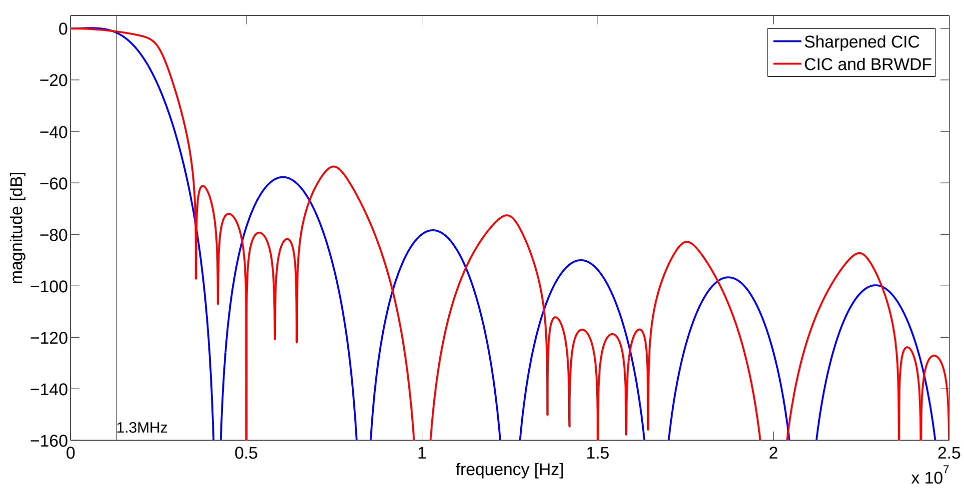


Figure 6.2: Comparison of frequency response

In figure 6.3 the group delay of both filters is depicted. The Sharpened CIC filter shows constant group delay whereas the CIC and Bi-Reciprocal filter introduces non-linear group delay. It is shows that the band of interest is not placed within the non-linear region and therefore distortions are not introduced.

6.3 SNR Simulations

The following SNR simulations are again done with the help of spectral analysis under consideration of coherent sampling as described in section 4.2.2. For the first result only

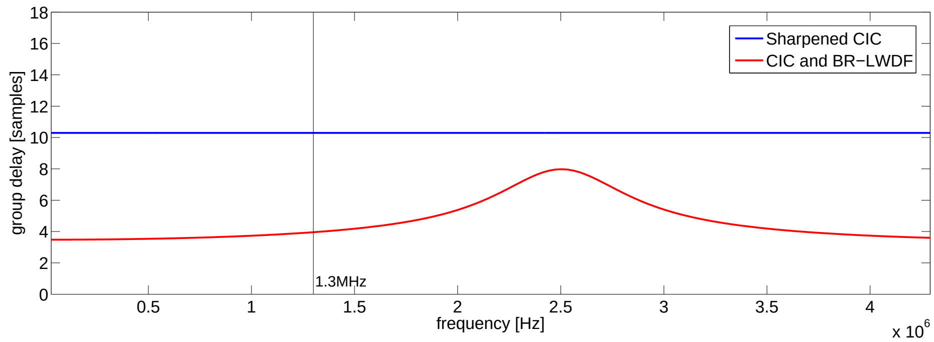


Figure 6.3: Comparison of group delay

the frequency band from zero to 1.3 MHz is observed. This makes the two implementations with different sampling frequencies comparable. The conclusion drawn from this simulation is that the Sharpening CIC filter performs slightly better than the other implementation for frequencies within the channel band (250 kHz to 750 kHz). The 1 300 kHz curve is identical for both filters. Furthermore a significant decrease of the SNR is observed at this frequency which is mainly caused by the quantization noise of the ΣΔ A/D converter as well as the passband droop of the CIC filter.

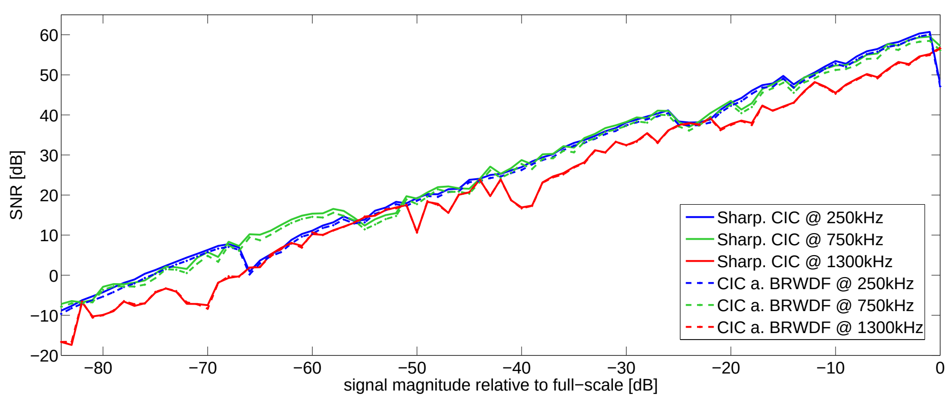


Figure 6.4: SNR simulations within 1.3 MHz frequency band

The SNR analysis with the reference channel filter introduced in section 4.1.2 shows satisfying results. Three frequencies were chosen within the channel band. The results show that after channel filtering the signal has 84 dB dynamic range with full scale input at the A/D converter. Even for input signals with -72 dB_{FS}, which equals a resolution of 12 bits, the SNR is still higher than 15 dB.

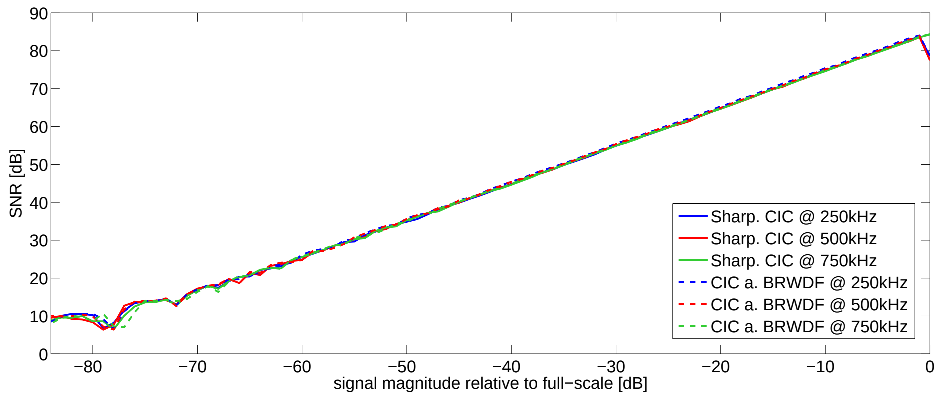


Figure 6.5: SNR simulations with channel filter

6.4 Area and Power Estimations

In order to obtain an absolute area estimation of the implemented decimation filters their VHDL code is synthesized with the Infineon C11N Platform Technology which is described in detail in [1]. It is a 130 nm CMOS technology for building logic, SRAM and analog/mixed signal applications and should be used in the design flow for the linear receiver.
In table 6.2 the synthesized chip area usage of both filter implementations is compared. The absolute values are declared in μm^2, the amount of combinatorial and non-combinatorial logic within one implementation is defined in percent.

Implementation	Module	Chip area [μm^2]	Non-Comb. [%]	Comb. [%]
Sharpened CIC	CIC	19 584		
	Sharpening	19 232		
	Sum	38 816	35%	65%
CIC and BR-LWDF	CIC	16 315		
	BR-LWDF	12 043		
	Sum	28 358	33%	67%

Table 6.2: Comparison of chip area usage

The results show that the CIC and Bi-Reciprocal LWDF is about a quarter smaller than the Sharpened CIC filter implementation. Especially the Bi-Reciprocal LWDF shows very low area usage although two coefficient multiplications are implemented. Obviously the Horner scheme for coefficient implementation is very efficient. By comparing the two CIC filter implementations the impact of decimation rate on the filter size can be observed. The sharpening structure occupies approximately the same amount of chip area as the CIC filter itself. Compared to the Bi-Reciprocal LWDF the Sharpening filter requires nearly 60% more chip area.

For an appropriate power estimation the number of registers and adders used in the filter implementation are weighted with the clock rates they are running at. As a common reference point the high frequency of the A/D converter (50 MHz) is used. Registers and adders within the 50 MHz clock domain have a power index of one. Thus components at lower clock rates have an index less than one. For example a flipflop, running on a clock rate which was decimated by five, has the power index of 0.2 only. In the end all values are summed up in order to compare both implementations in terms of power consumption. Table 6.3 shows the results of this method with power indices for registers and adders separately because these two cells do not consume the same amount of power and therefore cannot be mixed up.

Implementation	Module	Flip-flops [Power idx]	Adders [Power idx]
Sharpened CIC	CIC	120.58	120.58
	Sharpening	12	20
	Sum	132.58	140.58
CIC and BR-LWDF	CIC	102	102
	BR-LWDF	6.4	21
	Sum	108.4	123

Table 6.3: Comparison of power consumption with the help of power indices

In general it can be stated that the CIC and Bi-Reciprocal LWDF performs better because it has less flip-flops as well as full-adders running on the highest clock rate. This influences the power consumption most. Due to the high decimation in front of the BR-LWDF this filter shows very good results. But also the Sharpening filter is quite efficient when recalling it's relatively high amount of hardware components. But due to the low clock rate the filter is running at, power consumption is kept low.

Chapter 7

Conclusion

7.1 Summary

In this work two decimation filters for a linear wireless receiver were designed and implemented in VHDL. Great effort was put into finding a small and power efficient solution. Several filter structures were discussed and their advantages and disadvantages were presented. Conventional filter design methods for FIR filters showed up with a huge amount of coefficients. In contrast, IIR filters can be built with much less coefficients but they have a lot of drawbacks like non-linear phase, non-inherent stability, limit cycles and high sensitivity to coefficient quantization.
Another investigated decimation filter is the Cascaded Integrator Comb filter that has many advantages like adjustable decimation factor, low hardware effort, no coefficients and linear phase. But as a drawback it introduces strong passband droop. Compared to CIC filters, half-band filters are suitable for a rate decimation of two only. But as an advantage half of their coefficients are zero and can be neglected. This reduces the implementation size as well as power consumption. Half-band filters can be built as FIR and IIR systems.
Wave Digital Filters are IIR filters with favorable properties like guaranteed stability and insensitivity to coefficient quantization. These desirable properties make them an excellent choice for custom designs where size and power consumption matters. The most efficient structure is the half-band implementation called Bi-Reciprocal LWDF.

The first decimation filter implementation was a CIC filter with sharpening structure. It reduced the sampling frequency by a factor of 12. Due to the fact that CIC filters are efficient decimation filters but introduce heavy passband droop, a sharpening mechanism was added. Although the sharpening filter appeared to be plain, its structure turned out to occupy quite a lot of area. Nevertheless it has some unique advantages like linear phase and guaranteed stability due to it's FIR nature.
The other implementation was a CIC filter in cascade with a Bi-Reciprocal LWDF. The decimation factor of 10 was split up to 5 and 2. First the CIC filter decimated by five in order to avoid too much passband droop. Then the Bi-Reciprocal LWDF did additional attenuation of the CIC filter's main lobe. Because of the WDF's non-linear phase, higher decimation could not be done.

Filter verifications showed that both implementations perform more or less equal. In the channel band the Sharpened CIC filter was slightly better than the other implementation. But for higher frequencies the simulated results were exactly equal. The SNR simulations with additional channel filter showed that signals within the channel band have a SNR of at least 15 dB for an input signal of -72 dB_{FS}.
Concerning hardware cost the CIC and Bi-Reciprocal LWDF is about a quarter smaller compared to the Sharpened CIC filter according to the synthesis tool that reported the size for a 130 nm process technology. This is also reflected in the filter's list of used flip-flops and adders.
Due to the bigger size and the large amount of components operating at high clock rates, the power consumption of the Sharpened CIC filter is higher compared to that of the CIC and Bi-Reciprocal LWDF. It has to be concluded that mainly the efficient structure of the Bi-Reciprocal LWDF in combination with the coefficient multiplication done with Horner's algorithm are responsible for the small chip size and the low power consumption.

7.2 Outlook

Topics that were not covered in this work are mentioned now in this section.

One point that has not been investigated in the discussion about decimation filters is phase equalization of filters with non-linear phase. Higher decimation factors could be implemented in IIR filters if their phase response would not introduce distortion near the cut-off frequency. For example a cascade of four Bi-Reciprocal LWDF filters with a total decimation factor of 16 would be imaginable if their phase were linear. This topic has already been discussed in [24] and [17] in more detail.

In terms of filter verification there is one aspect that was not covered in the simulations. Although the decimation filters were implemented with the bandwidth for Direct Sequence Spread Spectrum (DSSS) in mind no evaluation has been done for this technique. Different implementation factors like aliasing and passband droop at frequencies around 1.3 MHz could negatively influence the results. Also the higher noise of the ADC at frequencies larger than 750 kHz contribute to a degradation of the signal. But as DSSS uses a wider frequency band for transmission, these influences may not be relevant.

The power estimation technique in section 6.4 represents a power index which is used for comparison between the two implementations only. For more precise results power simulations could be done with software packages like *PrimeTime*, developed by *Synopsys*. The power consumption of the implemented filters could be evaluated for different scenarios. But for this task a finished layout of the digital circuit is required.

The implemented filters were synthesized in order to obtain area estimations for a 130 nm CMOS technology. The next steps for verification of the functionality in the process technology would be a static timing analysis where signal delays dependent on place and route results are extracted.

Bibliography

[1] Infineon Technologies AG. C11N 130 nm CMOS Platform Technology. Technical report, Infineon, 2007.

[2] F. Corrêa Alegria and A. Cruz Serra. Coherent sampling of sinewaves. 2003.

[3] Ing. H.J. Lincklaen Arriëns. *(L)WDF Toolbox for Matlab*, 2006.

[4] P. M. Aziz, H. V. Sorensen, and J. vn der Spiegel. An overview of sigma-delta converters. 13(1):61–84, 1996.

[5] W. R. Bennett. Spectra of quantized signals. *Bell System Technical Journal*, 27:446–471, 1948.

[6] G. J. Dolecek and S. K. Mitra. Simple method for compensation of CIC decimation filter. *Electronics Letters*, 44(19):1162–1163, 2008.

[7] Anna Engelbert and Carl Hallqvist. Digital filters: Computable efficient recursive filters. Master's thesis, Chalmers University of Technology, 2008.

[8] A. Fettweis. Wave digital filters: Theory and practice. 74(2):270–327, 1986.

[9] L. Gazsi. Explicit formulas for lattice wave digital filters. 32(1):68–88, 1985.

[10] E. Hogenauer. An economical class of digital filters for decimation and interpolation. 29(2):155–162, 1981.

[11] W.Martin Snelgrove James A. Cherry, editor. *Continuous-time delta-sigma modulators for high-speed A/D conversion*. Kluwer Academic Publishers, 2000.

[12] G. Jovanovic-Dolecek and S. K. Mitra. Efficient sharpening of CIC decimation filter. In *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing (ICASSP '03)*, volume 6, 2003.

[13] Gordana Jovanovic-Dolecek. *Multirate Systems: Design and Applications*. Idea Group Publishing, 2002.

[14] J. Kaiser and R. Hamming. Sharpening the response of a symmetric nonrecursive filter by multiple use of the same filter. 25(5):415–422, 1977.

[15] Walt Kester. Taking the mystery out of the infamous formula, "snr = 6.02n + 1.76db," and why you should care. Technical report, Analog Devices, 2008.

[16] Scott D. Kulchycki. Continuous-time sigma-delta ADCs. Technical report, National Semiconductor, 2008.

[17] I. Kunold. Linear phase realization of wave digital lattice filters. In *Proc. Int Acoustics, Speech, and Signal Processing ICASSP-88. Conf*, pages 1455–1458, 1988.

[18] Richard G. Lyons. *Understanding Digital Signal Processing Second Edition*. Prentice Hall PTR, 2004.

[19] Reinder Nouta. *Studies in Wave Digital Filter theory and design*. PhD thesis, University of Technology Delft, 1979.

[20] Henrik Ohlsson. *Studies on design and implementation of low-complexity digital filters*. PhD thesis, University of Linköping, 2005.

[21] T. Parks and J. McClellan. A program for the design of linear phase finite impulse response digital filters. 20(3):195–199, 1972.

[22] Dietrich Schlichthärle. *Digital Filters: Basics and Design*. Springer, 2000.

[23] Mario Steinert. Implementierung und Anwendung einer Filterverikationsmethodik für den Entwurf von Digitalfiltern zur Untersuchung von Quantisierungseffekten. Master's thesis, Hochschule Mittweida (FH), 2000.

[24] A. Wroblewski, T. Erl, and J. A. Nossek. Bireciprocal lattice wave digital filters with almost linear phase response. In *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing (ICASSP '03)*, volume 2, 2003.

[25] A. Wroblewski and J. A. Nossek. Filter structures for decimation: a comparison. In *Proc. Int. Symp. Circuits and Systems ISCAS '03*, volume 4, 2003.