# University of Twente

Faculty of Electrical Engineering, Mathematics & Computer Science



## Increasing the Spurious-Free Dynamic Range of an Integrated Spectrum Analyzer

M.S. Oude Alink MSc. Thesis November 2008

**Supervisors** 

prof. dr. ir. B. Nauta prof. dr. ir. G.J.M. Smit dr. ir. A.B.J. Kokkeler dr. ing. E.A.M. Klumperink ir. K.C. Rovers

Report number: 067.3277a Chair of Integrated Circuit Design Chair of Computer Architecture for Embedded Systems Faculty of Electrical Engineering, Mathematics and Computer Science University of Twente P.O. Box 217 7500 AE Enschede The Netherlands

## Abstract

Spectrum Analyzers (SAs) are measurement instruments able to decompose a time signal into its frequency components. Due to non-idealities, SAs add noise and distort the signal to be measured. The ratio between the the largest signal and the noise floor level in a measured spectrum, without any distortion components rising above the noise floor, is called the Spurious-Free Dynamic Range (SFDR).

In a CMOS-integrated SA the SFDR is limited to around 60 dB by technology, while it needs to be 70 dB (at a frequency resolution of 1 MHz) to be competitive with commercial SAs. A method called crosscorrelation is introduced to lower the noise floor at the cost of measurement time. It relies on two equivalent measurement paths in which the noise produced in one path is uncorrelated with the noise produced in the other path, such that the noise in the final spectrum tends to cancel out. Although the noise level is only lowered by 1.5 dB if measurement time is doubled, it allows the SA to be designed for high linearity.

This design involves the use of digital hardware to compute the crosscorrelation. Consequently Analog-to-Digital Converters (ADCs) are required, but they also limit the SFDR due to the non-linear effect of quantization. New approximations to the relation between the number of quantization levels and the SFDR are found. These approximations show that every additional bit improves the SFDR by 8 dB. A simulator of the Montium 2 processor, which is still under development, is used to implement the digital correlation. Its fixed-point arithmetic proves sufficient for an SFDR of 87 dB.

An RF-frontend with a frequency range of 0 GHz to 6 GHz is designed for maximum linearity by moving amplification to IF. It provides impedance matching, variable attenuation and mixing. Its performance figures are a Noise Figure (NF) of 14 dB and a Third Order Input-referred Intermodulation Intercept Point (IP3) of +23 dBm, which gives a theoretical SFDR of 82 dB.

In order to obtain estimates on the feasability of an integrated SA, other parts, such as the IF-circuitry and local oscillators, are briefly reviewed. The estimated power consumption of the entire correlation SA is 0.5 W at a sample rate of 200 MS/s, and the estimated chip area is  $6.5 \text{ mm}^2$ . The largest power consumers are the VCO (0.2 W), followed by the IF-circuitry (0.1 W) and the ADCs and digital correlator (each 0.08 W). Chip area is dominated by SRAM-memory (36%), ADCs (25%) and the VCO (20%).

## Preface

I found it very hard to make a decision regarding the research topic for my master's thesis. The combination of electrical engineering and computer science provided plenty of opportunities. The open questions, the broad scope and the relatively good match with the courses I had finished finally made me decide to go for this assignment. During the year I spent on this project I learned that I really like doing research, whether it is the mathematical modeling, the interaction between parts of the system, the design of an analog integrated circuit or the implementation of a signal processing algorithm. I actually like it so much that for the next four years I will be working on a related topic as a PhD-student.

Doing research, however, is not the only important thing in a master's assignment; in my opinion, social contact on the workfloor is just as important. Spending time in both the CAES-group and the ICD-group, I liked the fact that in both groups the doors are always open, and one immediately feels at home. Coffee breaks were the time to put one's mind to more important matters. I would like to thank everyone from CAES, ICD and SC for the good atmosphere, the many laughs and the great time I had spending on their floors.

This thesis would not have been here in its present form without the help of many people, which I'd like to mention personally. First of all, I would like to thank my daily supervisors Eric Klumperink, André Kokkeler and Kenneth Rovers for a year of fruitful discussions and interesting ideas. The scheduled meetings and discussions with Gerard Smit and Bram Nauta helped a lot in structuring all material in a project where one could easily lose himself for a lifetime.

Marcel van de Burgwal was a great help in getting power estimates from VHDL-code. I would not have been able to run the circuit-level simulations without the assistance from Michiel Soer and Paul Geraedts. Xiang Gao provided valuable feedback and references on the oscillator design. This thesis would not have looked the way it does now without the help and advice regarding IATEX offered by Philip Hölzenspies and Pascal Wolkotte. Niels Moseley pointed out some references on signal processing which proved to be very valuable. Jordy Potman from Recore Systems is thanked for answering all my questions regarding the Montium 2. Many spelling and grammatical errors and unclarities have been picked out by my good friends Alfons Groenland and Regina Cadillac; any remaining errors are only mine to blame. I would also like to thank Fabian van Houwelingen, Karel Walters, Ed van Tuijl and Jasper Vrielink for their help and advice on different matters.

Without my parents Adrie and Hans I would never have made it this far. No words can express my eternal gratitude for giving me all the support I needed, and more.

Enschede, November 14, 2008

Mark Oude Alink

As our circle of knowledge expands, so does the circumference of darkness surrounding it.

—Albert Einstein [1879–1955]

## Contents

 $\mathbf{vii}$ 

Contents
----------

1	Intr	roduction 1			
	1.1	Spectrum analyzers			
	1.2	Integrated circuits			
	1.3	Applications			
	1.4	Previous research			
	1.5	Project description			
	1.6	Thesis outline			
ე	Correlation				
4	2.1	Introduction 0			
	2.1 0.0	Completion functions			
	2.2 0.2	The greateners 11			
	2.3 9.4	The spectrum			
	2.4	Correlation function estimation			
	2.5	Spectral estimation techniques			
	2.6	Spectral estimation			
	2.7	Correlation in a spectrum analyzer			
	2.8	Conclusions			
	2.9	Recommendations			
3	Quantization 29				
	3.1	Introduction			
	3.2	Quantization of a sinusoid			
	3.3	Mathematical derivation of the trend			
	3.4	Quantization of a sinusoid with noise			
	3.5	Multitone quantization			
	3.6	Example			
	3.7	Practical considerations			
	3.8	Conclusions			
	3.9	Recommendations			
1	Design of the spectrum analyzer 41				
4	1 1	System design 41			
	4.1	Coscording suppression mechanisms			
	4.2 19	Cascaling suppression mechanisms			
	4.0	rower consumption and cmp area			
	4.4	Measurement time			
	4 5				
	4.5	Conclusions			

CONTENTS
----------

5	Analog frontend5.1Attenuator5.2Circuit implementation5.3Simulation results5.4Two attenuators5.5SFDR of analog frontend5.6Comparison with other spectrum analyzers5.7Conclusions5.8Recommendations	<b>59</b> 59 64 66 74 80 80 82 82	
6	Digital Backend6.1Digital correlators6.2Hardware architectures6.3Design of an FX-correlator6.4Montium 26.5Mapping the FX-correlator onto the Montium 26.6Conclusions6.7Recommendations	<b>85</b> 94 96 98 100 109 112	
7	Summary & Conclusions         7.1       Conclusions         7.2       Future Research	<b>117</b> 119 120	
B	Derivations         B.1 Expectation of ccf estimator         B.2 Covariance of ccf estimator         B.3 Expectation of cross-spectrum estimators         B.4 Variance of cross-spectrum estimator         B.5 Asymptotic properties of SAVG         B.6 Asymptotic properties of XSA         B.7 Oscillator power         B.8 Input impedance of RF-frontend         B.9 Noise Figure of a Tayloe mixer         B.10 Noise Figure of RF-frontend         B.11 Algorithmic complexity of complex FFT	<b>125</b> 125 125 126 127 129 130 133 133 134 135 136	
С	Low Power VCO Idea	137	
D	Stochastic Processes	139	
Bi	Bibliography		

viii

#### Chapter 1

## Introduction

Sines and cosines are well-known mathematical functions. They have the special property that after a certain interval they repeat themselves, i.e. these functions are periodic. This interval is generally referred to as the period, and the inverse of the period is known as the frequency.

This notion of periodicity became especially important when in 1807 Baron Jean Baptiste Joseph Fourier (1768–1830) had written down his idea that an arbitrary function could be represented by a(n) (in)finite sum of sines and cosines, which he used to solve heat equations [1]. This possibly infinite series is commonly known as the Fourier series. It can be regarded as a method to convert a function from the time-domain to the frequency-domain.<sup>1</sup> Since then, many closely related mathematical operations or transforms were introduced, such as the continuous-time Fourier transform or the Discrete Hartley Transform. In short, Fourier theory describes the conversion between the discrete or continuous time domain and the frequency domain.

Periodicities play an important role in economical, geological and technical disciplines. Finding those periodicities or frequencies in a stream of data is the area of spectrum analysis. Knowing periodicities can be very important and has many applications, which can be observed from the following examples.

- Economic activity about a long-term growth trend appears to follow a superposition of so called *business cycles* of different length, ranging from 3 to 60 years [2]. This knowledge is used by central banks to 'stabilize' the economy and prevent another Great Depression.
- Knowledge of the frequency or frequencies of undesired mechanical vibrations in machinery can be used to find the cause or to derive appropriate measures to mitigate them.
- In many applications of telecommunications, several transmitters are sending data at the same time but at different frequencies. Authorities governing the spectrum dictate which frequencies and under which circumstances one is allowed to transmit, and spectrum analysis can indicate whether a device is adhering to these standards.
- Music contains frequencies ranging from roughly 0–20 kHz. Many people cannot really hear any difference when frequencies close to 20 kHz are removed, and this fact is used to compress music to create MP3-files.
- Spectral measurements of the light received from stars indicate their velocity and direction of movement, the elements they are composed of and their surface temperature.

<sup>&</sup>lt;sup>1</sup>The terms *time domain* and *frequency domain* are used even though it is not necessarily limited to a period in seconds or a frequency in Hz; it might as well be a period in meters and a frequency in  $m^{-1}$  or any other quantity for that matter.



Figure 1.1: Approximation of a periodic square wave with a (truncated) Fourier series.



Figure 1.2: Another way of looking at Fourier theory is that it relates two different perspectives, time and frequency, on the same object (reproduced from [3]).

#### 1.1 Spectrum analyzers

A Spectrum Analyzer  $(SA)^2$  is a device able to determine the spectral contents of a time domain signal. An SA is limited to a certain frequency range, and this frequency range determines its architecture. An SA for visible light ( $f \approx 300 \text{ THz} - 800 \text{ THz}$ ) has a structure completely different from an SA for audible sound ( $f \approx 20 \text{ Hz} - 20 \text{ kHz}$ ). SAs usually work on Electromagnetic (EM) input signals, which include for example radio waves, X-rays and light (see fig. 1.3). Determining the spectral contents of other types of signals is mostly done by first converting the signal to the electrical domain and then analyze it using an SA. In the case of audio this function can be performed by a microphone, while light can be converted to the electrical domain by means of an optical diode.

Nowadays, numerous manufacturers offer commercially available SAs, each designed to excel in one or more areas. The specifications of high-end SAs are really good in terms of linearity and absolute amplitude accuracy for instance, but not every application requires such high-quality information. Of course, having excellent specifications comes at a high price (in the order of  $\leq 10,000$  to  $\leq 100,000$  [4]), making the high-end systems far too expensive for many applications. Furthermore, these high-end systems are often bulky and consume a lot of power, making measurements outside of the lab difficult. To accommodate these discrepancies, handheld and budget SAs are also readily available. Coming with lower

 $<sup>^2{\</sup>rm The}$  British-English spelling is 'spectrum analyser'; this thesis adheres to American-English.



Figure 1.3: The RF-range is part of the EM-spectrum.



Figure 1.4: Two commercial SAs

specifications, prices still range from  $\in 1,000$  to  $\in 10,000$  [4], which prevents their use where many of them are required.

The price is mostly set by the high requirements of the components used inside of an SA. These high requirements demand the use of exotic and expensive techniques, which do not benefit from the tremendous progress made in the mass production of mainly digital electronics.

#### **1.2** Integrated circuits

Digital electronics are usually made out of Metal-Oxide-Semiconductor (MOS)-transistors. A combination of n-type MOS (NMOS) and p-type MOS (PMOS)-transistors, also referred to as Complementary MOS (CMOS)-technology, allows a lot of circuitry to be put into a small area with a low power consumption. Because it is so widely used and so much effort has been put into research, the price of a single transistor has gone down by a factor of ten million in the last forty years [5].

Analog electronics designers often try to take advantage of this cheap production process, even though CMOS is optimized for digital circuits. Some disadvantages of CMOS for analog circuits include:<sup>3</sup>

 $<sup>^{3}</sup>$ It is interesting to note that with the ever-decreasing size and operating voltage of CMOS-transistors, digital circuitry is starting to suffer from the same disadvantages, as it is inherently analog [6, 7].



Figure 1.5: Power consumption of analog and digital CMOS implementation as a function of the SNR (adapted from [9]).

- Spread in component sizes and doping levels, which changes the characteristics of the components in a statistical sense;
- A large threshold voltage, decreasing the usable voltage range;
- Low-Q components (capacitors and inductors with a large resistance), which makes it very difficult to make highly selective filters;
- A noise level higher than in e.g. bipolar circuits, which decreases attainable specifications;
- Relatively low speed, which makes it unsuitable for certain high-frequency applications (although progress is being made in this area).

These disadvantages have led designers to explore other architectures for 'old' problems [8]. One very good example is the  $\Sigma\Delta$ -modulator, which can be used in Analog-to-Digital Converters (ADCs) or Digital-to-Analog Converters (DACs). In many ADC and DAC architectures, the matching of components<sup>4</sup> limits the attainable specifications. Better matching requires less relative variation of component parameters in the production process. Variation can be reduced by increasing the size of the components, but this increases power consumption and reduces the speed because of larger capacitances. The  $\Sigma\Delta$ -modulator reduces matching requirements by using quantizers with only a few quantization levels. By means of oversampling, noise shaping and digital filtering, very high specifications can be obtained. Besides using the digital capacities of CMOS for filtering, oversampling requires a high switching speed, something at which CMOS is also rather good.

Because digital circuits are so cheap nowadays, more and more parts of SAs are digital. Nature is still analog though, so an analog frontend will always be necessary. As both analog and digital circuits can be fabricated in the same process, they can be combined to form mixed-signal circuits. This naturally raises the question of what parts should be made analog and what parts digital.

In analog circuits one is usually interested in the Signal-to-Noise Ratio (SNR), which defines the ratio between the power of the desired signal and the noise. In digital circuits the SNR is indirectly defined by the number of bits used, so a natural way of comparing analog and digital implementations would be to determine the amount of power required to get a certain SNR, and select the implementation that requires the least power. This approach has led to the graph shown in fig. 1.5.

Physical considerations lead to a fundamental limit on the analog power consumption, while practical circuits currently require about three orders of magnitude more in power.

<sup>&</sup>lt;sup>4</sup>The degree to which components can be made equal.

#### 1.3. APPLICATIONS

Doubling the SNR (increasing it by 3 dB) requires twice the amount of power for an analog system. In the digital domain, however, each extra bit gives an increase of 6 dB in SNR, while requiring only a little extra power. The break-even point moves to lower SNR as newer technologies tend to decrease the power consumption required by digital circuits, but have (almost) no effect on analog circuits.

It is important to note that the graph does not take into account the power required to transform a signal from the analog to the digital domain or vice versa. Constantly switching from one domain to the other is definitely not better, and therefore the trade-off is application-specific.

Another question is whether it is possible to let the analog or digital part solve some issues of the other part. This is still an active area of research, and will also be touched upon in this thesis.

#### **1.3** Applications

There is no doubt that an all-CMOS, and therefore cheap, implementation of an SA is possible as [10–12] show. In fact, patents have already been filed [13, 14]. The question is, what kind of specifications will be attainable if the frequency range is extended to higher frequencies to include applications like FM-radio ( $f \approx 100$  MHz), GSM-phones ( $f \approx 1$  GHz), Wireless LAN, Bluetooth ( $f \approx 2.4$  GHz) and HiperLAN ( $f \approx 5.8$  GHz), or even further up into the microwave region. With good specifications, such an SA would have many 'new' applications, such as:

- Cognitive Radio (CR). CR is a technique under development to utilize unused parts of the spectrum, for example to allow better and more reliable communication between emergency services, which currently rely on public networks [15, 16]. In order to know which parts of the spectrum are unused, an SA integrated with the device is required. As most of the communication will be through mobile devices, the power consumption and size of the SA needs to be kept low, while at the same time it should still be able to detect modern noise-like digital communication signals.
- instrument tuning. With the use of an SA, musical instruments can be tuned to any setting without the need for tuning forks or hiring a professional instrument tuner. In fact, the instrument may be able to tune itself with the addition of some tuning mechanism. Gibson introduced such a guitar, the *Robot Guitar*, in December 2007, because "every music lover and performer has had to suffer through the showhalting, mood-killing atonal droning of a loudly amped guitar being brought into tune",<sup>5</sup> with an additional cost of \$900.<sup>6</sup>
- Built-In Self-Testing (BIST). Circuits do not only suffer from degradation, but circuit parameters also drift over time. A built-in SA would be able to detect, and with some control circuitry perhaps even correct, these drifts. It might also give warnings or shut down (part of) the system in case certain (safety) limits are exceeded.
- measuring on-chip signals. This application is closely related to BIST. The idea is to use the SA to provide a common port to externally read out the spectrum of internal signals. Without this common port, every signal on the chip that needs to be measured at some point in time requires its own pin on the package. Moreover, the pins and associated bondwires can give problems at high frequencies because of parasitic inductance. Therefore, an internal SA may not only reduce the number of pins required on the package, but can also reduce measurement problems at high frequencies. The internal SA can process the internal analog or digital signals and

<sup>&</sup>lt;sup>5</sup>Gibson website: http://www.gibson.com/robotguitar/story.html

<sup>&</sup>lt;sup>6</sup>ABC News website: http://abcnews.go.com/Technology/wireStory?id=3949313

output them in digital form at a suitable rate. In both cases, the SA should be fully integrable and power-efficient.

• a PC plug-in card. With a computer as a digital signal processing backend, many plug-in components, including TV-tuners, ADCs, oscilloscopes and SAs, are available. They usually come in the form of a PCI-card or USB-peripheral. With an integrated SA, such a device would become affordable to more people.

#### 1.4 Previous research

In his Master's Thesis, Rovers [4] investigated the problems and limitations a wideband all-CMOS SA would have, where wideband should be interpreted as  $f \approx 0-3/6$  GHz. This large frequency range prohibits the use of direct Analog-to-Digital (AD)-conversion, as it is not yet technically possible. Even if it would have been possible, it would require (by extrapolating from current state-of-the-art designs) in the order of 2 W for 8-bit conversion at 10 GHz sampling rate [17], which is not low-power.<sup>7</sup> This limitation means that *frequency conversion* is necessary.

Frequency conversion is usually done by a mixer which multiplies the signal with a sine or block wave, such that the information that was present at some frequency is present at another frequency after the multiplication. The eventual goal is to get the desired frequencies low enough to be able to sample them. This can be done in one or multiple stages. In each stage the desired signal (which is a fraction of the total input band) is converted to some frequency, called the Intermediate Frequency (IF). Each IF is above, within or below the input signal range, or at 0 Hz. These are known as high-IF, mid-IF, low-IF and zero-IF respectively. Each of these architectures has advantages and disadvantages.

High-IF is the architecture often used in commercial SAs with a comparable frequency span [3, 19]. It has the advantage that RF-feedthrough<sup>8</sup> in the mixer does not interfere with the signal of interest, which is the main problem of mid-IF. The downside of high-IF is that a high-Q filter<sup>9</sup> is required to remove the images<sup>10</sup>, which is not feasible with a CMOS-implementation. For mid-IF the filter-Q requirement is less stringent than for high-IF, but in general it is still too high for a CMOS-implementation. Low-IF suffers from images that are originally close to the desired frequency, because before conversion the image is separated from the desired signal by only twice the IF. Filtering this image before conversion is not possible because it requires a high-Q filter, and the only way to get rid of them afterwards is to use a quadrature architecture.<sup>11</sup> Using a quadrature architecture, image rejection is typically limited to 40 dB because of mismatch, but may be increased to 60 dB [4]. Zero-IF does have less problems with images as the signal is its own image, but it suffers from DC-offsets (caused by self-mixing which distorts the signal and saturates subsequent stages), I/Q mismatch (corrupting the signal), even-order distortion and flicker noise<sup>12</sup>.

Analysis of the advantages and disadvantages of each of these architectures led Rovers to the conclusion that zero-IF and low-IF are the best options for full integration, which is in accordance with literature for integrated wideband receivers [17, 20–23]. In either case, the disadvantages and the practical limitations of CMOS limit the maximum difference between

<sup>&</sup>lt;sup>7</sup>The theoretical minimum power consumption for an 8-bit converter sampling at 10 GHz is around 10  $\mu$ W [18], which is five orders of magnitude lower than the extrapolation done by [17], consistent with observations made on practical implementations [18].

<sup>&</sup>lt;sup>8</sup>part of the signal directly appears at the output

 $<sup>^{9}</sup>$ The Q of a filter specifies the ratio between the center frequency of the passband and the width of this passband. Filters with higher Q are therefore better at removing signals close in frequency to the desired signal.

<sup>&</sup>lt;sup>10</sup>Images are signals at other frequencies that, because of the frequency conversion, are mapped to the same frequency as the desired signal.

 $<sup>^{11}</sup>$ A quadrature architecture splits the signal into two paths which are 90° out of phase. This allows the differentiation of positive and negative frequencies, and hence differentiation between desired signal and image.

<sup>&</sup>lt;sup>12</sup>low-frequency noise which can be a major issue in CMOS



Figure 1.6: The SFDR is limited by the noise floor and the non-linearity of the SA: (1) DC-term because of non-linearity, (2) signal lost in noise, (3) signal detected, (4) signal lost by non-linearity, (5) signal detected.

a larger signal and a smaller signal such that they can still both be observed. Increasing the power of the larger signal causes harmonics introduced by non-linearities in the circuit to obscure the smaller signal, while decreasing the power of the smaller signal makes it indistinguishable from the noise floor. This is visualized in fig. 1.6, where a clean signal consisting of four sinusoids of different amplitudes is processed by an SA. Two sinusoids are detected, but the weakest sinusoid is obscured by the noise, and one of the others is obscured by a harmonic of the strongest sinusoid.

This maximum difference, limited by the noise floor on one side and by linearity on the other side, is called the Spurious-Free Dynamic Range (SFDR), and is a very important property of SAs. The noise floor is lowered, thereby increasing the SFDR, if the Resolution Bandwidth (RBW) (the width of each frequency bin) is chosen smaller, because the total amount of noise power in each bin is proportional to its width in Hz. Commercial SAs typically have an SFDR of 70 dB for an RBW of 1 MHz, while for CMOS it is limited to roughly 60 dB [4].

#### 1.5 **Project description**

SAs are instruments used to scan a specific frequency range and output the measured power or amplitude (and possibly phase) into some form, usually a human-readable display. Commercially available SAs are expensive and, because of stringent requirements for different parts of the analyzer, not integrable on one chip.

An initial study on the front-end of an integrated low-cost SA showed that the SFDR is limited in standard CMOS-technology. Linearity can be improved at the cost of noise and vice versa. One solution proposed is the use of two measurement paths in which the signal is correlated and the noise is not, such that the noise may be removed in the digital domain and the system can be designed for high linearity.

The aim of this project is to:

- Explore the effects and possibilities using two measurement paths
- Explore the effect on SFDR of different blocks in the chain
- Design (part of) the analog frontend and digital backend to maximize SFDR in combination with two measurement paths
- Estimate the required power consumption and attainable specifications of the entire system and pinpoint the bottlenecks

#### 1.6 Thesis outline

The idea of this thesis is to break the tradeoff between noise and linearity using two measurement paths. Chapter 2 investigates the mathematical properties of this architecture with respect to spectral estimation, and compares it to the traditional approach using one measurement path. Because the crosscorrelation itself is done in the digital domain, an ADC is necessary. Correlation between the noise sources in both measurement paths will ultimately limit the reduction of the noise floor. Since an ADC also introduces 'noise' because the digital values do not exactly represent the analog values, the effects of quantization on SFDR are discussed in chapter 3.

Using the results from chapters 2 and 3, a high-level system design of a correlation SA is discussed in chapter 4. Power consumption and chip area of all the analog components are estimated. Because usually the RF-part of a receiver limits the linearity, an RF-frontend is designed and simulated in chapter 5 to investigate the maximum attainable linearity of a correlation SA. The implementation and simulation of the digital correlator is discussed in chapter 6, which also estimates the required power consumption and chip area.

In chapter 7, the results of all the preceding chapters are summarized, and an overview is given of the power and chip area requirements of all the different blocks in the correlation SA. It finishes with the most important topics for future research. Chapter 2

## Correlation

As explained in the introduction, the SFDR in the proposed SA is limited by the noise floor and linearity. Increasing linearity of analog components inadvertently means increasing the noise floor, while lowering the noise floor of components results in a decrease of linearity. For example, adding a Low-Noise Amplifier (LNA) in front of a highly linear but noisy mixer leads to a better Noise Figure (NF), but makes the nonlinearity contribution of the mixer larger. Therefore, the maximum dynamic range is limited by technology. In order to break this trade-off between the noise floor and linearity, another technique is needed.

One of the possibilities is the use of crosscorrelation, which will be investigated in this chapter. It will be shown that crosscorrelation has the effect of lowering the NF at the input of the SA, but at the cost of longer measurement time. It is important to note that the use of crosscorrelation to lower the noise floor of an SA is not new, see for example [24].

To compare this approach with that used by 'regular' SAs, which is shown to be related to crosscorrelation, the mathematical properties of crosscorrelation estimation will be discussed. Asymptotic expressions are derived using results found in literature, and compared to simulations, which show a good match. Based on practical limitations, crosscorrelation ultimately allows smaller signals to be detected than using the method of standard SAs.

This chapter involves the use of some signal theory, stochastic processes and statistics, of which the relevant basics and further references are covered in appendix D. Furthermore, some knowledge of Fourier theory is required to understand the line of reasoning (see for example [1, 25]).

#### 2.1 Introduction

Crosscorrelation of two signals gives information on the relation between the first signal with a time-delayed version of the second signal. Informally, the result of crosscorrelation is a function that shows the degree of resemblance between two signals as a function of the delay-time of the second signal.

Any measurement system adds noise to the signal to be measured, which can corrupt, and, in case the signal is very weak, even completely obscure the signal. Crosscorrelation has the property that if the noise sources in both signal-paths are uncorrelated, the noise tends to cancel out and only the crosscorrelation of the signal(s) remains.

This important property is used in arrays of radiotelescopes where signals are well below the noise floor. An additional advantage of crosscorrelation is that it also gives a phase relation, which allows one to calculate the position of the source. One practical aspect is that this correlation may go on for hours or even days to lower the noise floor to negligible levels. This was also the case in an experiment performed by Sampietro et al. [26] where the crosscorrelation technique was used to measure the thermal noise of a resistor. The measurement results are reproduced in fig. 2.1.



Figure 2.1: Measurement results of resistor noise performed by Sampietro et al. [26] shows that crosscorrelation works and that performance increases with measurement time. The one-channel measurement mimics a standard spectrum analyzer, where the internally generated noise obscures the resistor noise.

For some applications, such as radioastronomy, this long measurement time is not an issue, but for spectrum analysis it is. The correlation time completely depends on the specific application. Many signals have a finite duration, which automatically puts a constraint on the allowed measurement time.

#### 2.2 Correlation functions

The crosscovariance function of stochastic processes X(t) and Y(t) is defined as

$$\gamma_{XY}(t,\tau) \stackrel{\triangle}{=} E\left[\left(\overline{X(t)} - E\left[\overline{X(t)}\right]\right) \left(Y(t+\tau) - E\left[Y(t+\tau)\right]\right)\right]$$

where  $\overline{X(t)}$  denotes the complex conjugate of X(t). Note that some authors use a slightly different definition with respect to the complex conjugate or the delay, but this has no consequences to the general idea. The crosscorrelation function (ccf) is defined as

$$\rho_{XY}(t,\tau) \triangleq \frac{\gamma_{XY}(t,\tau)}{\sigma_X(t)\sigma_Y(t+\tau)}$$

which is just a scaled version of  $\gamma_{XY}$ . The terms 'covariance function' and 'correlation function' are sometimes used interchangeably, which may be because scaling is not necessary for spectral estimation (in fact,  $\sigma_X$  and  $\sigma_Y$  are generally not known in measurements and need to be estimated themselves). For now it is assumed that all signals have a mean value of 0; corrections for this will be given later on.

Because in general only one realization can be observed, the ccf of stochastic processes cannot be measured. Therefore, it is often assumed that the processes are *jointly ergodic* in the first and second order moments. The first consequence of having ergodic processes is that the first and second order moments do not change in time (i.e., they are wide-sense stationary (wss)). Hence the ccf is independent of absolute time:

$$\gamma_{XY}(\tau) \stackrel{\triangle}{=} E\left[\overline{X(t)}Y(t+\tau)\right] \tag{2.1}$$

which will be the definition of the ccf used in this document. Setting Y = X yields the autocorrelation function (acf) of X(t).

#### 2.3. THE SPECTRUM

The second consequence of being ergodic is that the time average of one realization equals the ensemble average:

$$\gamma_{XY}(\tau) = \lim_{T \to \infty} \frac{1}{2T} \int_{-T}^{T} \overline{x(t)} y(t+\tau) dt$$

This means an estimate of the ccf can be made by observing one realization for a certain amount of time.

It is easy to prove that  $|\gamma_{XX}(\tau)| \leq \gamma_{XX}(0)$  for all  $\tau$ , and  $\gamma_{XY}(-\tau) = \gamma_{YX}(\tau)$ . It is assumed that X(t) and Y(t) are real stochastic processes, and as a result any realization x(t) and y(t) is real-valued. This means that the acfs of X(t) and Y(t) and of any realization x(t) and y(t) are even functions.

#### 2.3 The spectrum

Determining the spectrum is the ultimate goal of an SA. Transforming a continuous time domain signal to a frequency domain signal can be done using the Continuous-Time Fourier Transform (CTFT). In literature several definitions are in use for the CTFT, each differing in the factors before the integral. In this thesis, the CTFT is defined as:

$$G(f) \stackrel{\triangle}{=} \mathfrak{F}(g(t)) \stackrel{\triangle}{=} \int_{-\infty}^{\infty} g(t) e^{-j2\pi ft} \,\mathrm{d}t$$
(2.2)

and its inverse is

$$g(t) \stackrel{\triangle}{=} \mathfrak{F}^{-1}(G(f)) \stackrel{\triangle}{=} \int_{-\infty}^{\infty} G(f) e^{j2\pi ft} \,\mathrm{d}f$$
(2.3)

which in radians (with  $\omega = 2\pi f$ ) becomes

$$g(t) = \mathfrak{F}^{-1}(G(\omega)) = \frac{1}{2\pi} \int_{-\infty}^{\infty} G(\omega) e^{j\omega t} \,\mathrm{d}\omega$$

The term  $|G(f)|^2$  should be interpreted as the Power Spectral Density (PSD) of the function g(t), i.e. the amount of power per Hertz at each frequency.

A discrete version of the CTFT also exists, which is known as the Discrete-Time Fourier Transform (DTFT). It is constructed from the CTFT by sampling the continuous-time signal, which in the frequency domain corresponds to a convolution with a series of  $\delta$ -pulses. The result is

$$G(f) \stackrel{\triangle}{=} \sum_{n = -\infty}^{\infty} g[n] e^{-j2\pi f n}$$

It immediately follows that G(f) is periodic in f with period 1.

As can be seen, the CTFT and DTFT range over infinite time. In a practical digital system, only a finite number of values are available, simply because both measurement time and sampling rate are limited. Therefore, it would be useful to be able to digitally compute (an approximation of) the CTFT and/or DTFT. This transformation is known as the Discrete Fourier Transform (DFT). The definition used in this thesis for the DFT is

$$G[k] \stackrel{\triangle}{=} \mathrm{DFT}\left(g\right) \stackrel{\triangle}{=} \sum_{n=0}^{N-1} g[n] e^{-j\frac{2\pi}{N}nk}$$
(2.4)

where g[n] denotes the *n*-th sample of the sequence g and N the length of the sequence g. Its inverse transform is

$$g[n] \stackrel{\triangle}{=} \text{IDFT}(G) \stackrel{\triangle}{=} \frac{1}{N} \sum_{k=0}^{N-1} G[k] e^{j\frac{2\pi}{N}nk}$$
(2.5)

An important property of autocorrelation is the fact that the power spectrum  $\Gamma_{XX}(f)$ and the acf  $\gamma_{XX}(\tau)$  form a CTFT pair, which is captured by the Wiener-Khinchin theorem [27, 28]<sup>1</sup>:

$$\Gamma_{XX}(f) = \int_{-\infty}^{\infty} \gamma_{XX}(\tau) e^{-j2\pi f\tau} d\tau$$

$$\gamma_{XX}(\tau) = \int_{-\infty}^{\infty} \Gamma_{XX}(f) e^{j2\pi f\tau} df$$
(2.6)

The function  $\Gamma_{XX}(f)$  has the following properties (note that it is assumed that X is real)

- $\Gamma_{XX}(-f) = \Gamma_{XX}(f)$
- $\Gamma_{XX}(0)$  is the DC-power of the process X
- $\Gamma_{XX}(f) \ge 0$

Fourier-transforming the ccf instead of the acf results in a so-called cross power spectrum (or simply cross-spectrum)  $\Gamma_{XY}(f)$ . The cross-spectrum is not necessarily a real function of frequency like the power spectrum is; in fact, it may not have any physical meaning at all (although it will be shown that the cross-spectrum will converge to a true power spectrum in the situation discussed in this thesis). The function  $\Gamma_{XY}(f)$  has the property  $\Gamma_{XY}(-f) = \overline{\Gamma_{YX}(f)}$ .

#### 2.4 Correlation function estimation

Estimating the correlation function is the first step towards estimating the spectrum, although for some applications other than spectrum analysis it may also be of interest by itself. The estimation can be done in either the analog or the digital domain. It requires delays, additions and multiplications. Because additions and multiplications add their own noise (which will not be correlated away due to the fact that the two paths have been combined at this point), only the digital domain implementation, and hence only the digital domain estimation process, is considered. This does not mean that analog correlation is not a good choice, see e.g. [31].

Two estimation functions widely used are

$$c_{XY}[k] = \frac{1}{N} \sum_{n=1}^{N} \overline{x[n]} y[n+k]$$
(2.7)

$$\hat{c}_{XY}[k] = \frac{1}{N - |k|} \sum_{n=1}^{N} \overline{x[n]} y[n+k]$$
(2.8)

where N is the total number of samples taken, k the lag (delay) in number of samples, and the summation is not carried out for terms which contain a sample that does not exist (for instance, for k = N - 1, the summation reduces to a single term). These estimators are used because of their intuitive appeal, not because they possess the best mathematical properties. Determining the optimal estimator proves to be intractable [32].

A third estimator, which is often used in radioastronomy, is very similar to eq. (2.7). Instead of directly starting the correlation process at the first sample, one first waits until a sample is available for all desired lags. It has the advantage that each lag is calculated using the same number of samples, which makes all lags equally reliable. Other advantages are that each lag can be divided by the same number, and that the digital cells doing the

<sup>&</sup>lt;sup>1</sup>Apparently, Albert Einstein already derived these equations, albeit without a rigorous mathematical proof [29, 30].

#### 2.4. CORRELATION FUNCTION ESTIMATION

calculation can be switched on and off at the same time. The main disadvantage is that not all available information is used, but that effect is negligible in radioastronomy where measurement times are very long. The properties of this estimator are not further explored.

 $\hat{c}_{XY}[k]$  is an unbiased estimator as it divides by the exact amount of overlapping samples for a given lag k. However, the variance grows to such high values for k close to N, that the mean-squared error (mse) is larger than the mse of  $c_{XY}[k]$ , even though the latter is biased. We will therefore restrict ourselves to this biased estimator. It will turn out that for spectral estimation both estimators can be treated in exactly the same way, which makes this choice justified.

Converting the results derived in [32] from continuous- to discrete-time (see sections B.1 and B.2), the expected value of  $c_{XY}[k]$  is

$$E[c_{XY}[k]] = \left(1 - \frac{|k|}{N}\right)\gamma_{XY}[k]$$
(2.9)

and its variance is

$$\operatorname{var}\left(c_{XY}[k]\right) = \frac{N - |k|}{N^2} \sum_{n = -(N - |k|)}^{N - |k|} \left(1 - \frac{|n|}{N - |k|}\right) \times \left(\gamma_{XX}[n]\gamma_{YY}[n] + \gamma_{XY}[n + k]\gamma_{YX}[n - k]\right) \quad (2.10)$$

such that the bias is

$$B[c_{XY}[k]] = \frac{|k|}{N} \gamma_{XY}[k] \tag{2.11}$$

Clearly, the estimator is consistent and asymptotically unbiased.

#### 2.4.1 Correlation and convolution

It is possible to rewrite the biased estimator to another familiar and useful form. Defining discrete convolution as

$$(f * g)[m] \stackrel{ riangle}{=} \sum_{n} f[n]g[m-n]$$

and discrete correlation as

$$(f \star g)[m] \stackrel{ riangle}{=} \sum_{n} \overline{f[n]} g[n+m]$$

it immediately follows

$$f[n] \star g[n] = \overline{f[-n]} * g[n] \tag{2.12}$$

Note that in the above definitions constant factors were omitted. Since both operations are linear, they can easily be incorporated.

#### 2.4.2 Non-zero mean

So far the possibility of the process having a mean unequal to zero was neglected. In the general case a non-zero mean value will have an influence on the estimation.

The ccf defined in eq. (2.1) is a simplified version where it is already taken into account that the mean is zero. The adaptation for the general case is straightforward:

$$\gamma_{XY}(\tau) \stackrel{\triangle}{=} E\left[ (\overline{X(t)} - E\left[\overline{X(t)}\right])(Y(t+\tau) - E\left[Y(t+\tau)\right]) \right]$$

When the mean is known beforehand, it can simply be subtracted from the data of an observation and all the previous results still hold. When the mean is unknown, it needs

to be estimated before it can be subtracted. The natural and most used estimator for the mean is

$$M_X = \frac{1}{N} \sum_{n=0}^{N-1} X[n]$$

It can be shown to be unbiased and having a variance of  $\operatorname{var}(X)/N$  [32].

Plugging in this estimate into  $c_{XY}[k]$  changes the amount of bias of  $c_{XY}[k]$  to [32]:

$$B[c_{XY}[k]] = \frac{|k|}{N} \gamma_{XY}[k] + \left(\frac{1}{N} - \frac{|k|}{N^2}\right) \sum_{n=-(N-1)}^{N-1} \left(1 - \frac{|n|}{N}\right) \gamma_{XY}[n]$$

For virtually all statistical processes,  $\gamma_{XY}(\tau)$  will be close to 0 for  $|\tau| > \tau_0$ , where  $\tau_0$  is some constant. Therefore, if the number of samples is high enough, this extra bias can be neglected.

#### 2.5 Spectral estimation techniques

For an SA it is of the utmost importance that the estimates it makes are statistically meaningful, i.e. that the resulting values have a certain degree of accuracy and repeatability. Over the years, many techniques for spectral estimation have been developed, each with its own advantages and disadvantages.

These techniques can be divided into three main categories: classic non-parametric estimation, parametric estimation and subspace spectral estimation. Some alternatives, such as using a neural network for spectral estimation [33, 34], wavelets [35] or non-uniform sampling [36–38], are not included in any of these categories and are not considered for implementation in this MSc project.

#### 2.5.1 Classic non-parametric estimation

Non-parametric estimation makes no assumption on the type of spectrum it tries to estimate. It generally uses some form of the DFT to go from the time domain to the frequency domain, and is relatively computationally efficient.

Because it makes no assumptions, it works reasonably well for a large class of signals. Disadvantages include a low frequency resolution, which makes it impossible to distinguish between two closely spaced signals, and spectral leakage, which causes power in some frequencies to 'leak' to other frequencies, thereby giving incorrect spectral values.

Spectral leakage is caused by time-windowing the sampled sequence in combination with the mathematical property of the DFT to assume the signal to be periodic in the given measurement time. The leakage can be reduced by using tapering windows which usually go to zero at the edges, such that the discontinuities disappear. Unfortunately, reducing spectral leakage comes at the cost of loss in frequency resolution, see fig. 2.2. Many windows exist for which spectral leakage and frequency resolution have been calculated and tabulated. The window should be chosen to suit the situation.

The frequency resolution in Hz is reciprocal to the time interval in seconds over which sampled data is available: a higher frequency resolution requires longer sampling and more points in the DFT. Because the Fast Fourier Transform (FFT), which is an efficient implementation of the DFT, has a computational complexity of  $O(n \log n)$  and a memory complexity of O(n) [25], there is a practical upper limit to the number of points in the FFT.

Longer sampling is not always an option, as some processes are simply of finite duration or have time-varying spectra.<sup>2</sup> In these cases, the non-parametric estimation methods can

 $<sup>^{2}</sup>$ A time-varying spectrum is a *contradictio in terminis* as a true spectrum is defined over infinite time. However, one can imagine that it is useful to measure the spectrum of a frequency-hopping protocol such as Bluetooth during the time it is using one of the frequency slots.



Figure 2.2: Windowing is a tradeoff between frequency resolution and spectral leakage.

prove insufficient, and one resorts to other types of estimation.

#### 2.5.2 Parametric estimation

In many situations one can estimate the values of the signals *outside* of the observed period because some properties of the signal to be measured are known. The use of *a priori* information allows better estimation of the spectrum.

In parametric estimation, a model with some parameters is used. This model is usually a linear system with frequency response H(f), where the parameters determine this frequency response. By using the equation  $\Gamma_{XX}(f) = |H(f)|^2 \Gamma_{WW}(f)$  (in which  $\Gamma_{WW}(f)$  is the power spectrum of white noise) the parameters are estimated by observing the signal. The estimated frequency or impulse response of the system provides all the information for the power spectrum estimation.

The main advantage of this way of estimation is the higher frequency resolution that can be obtained as compared to non-parametric estimation, because with parametric estimation the acf is not set to zero outside the measurement range. The obvious disadvantage is that a model is required that reflects the process well enough.

A commonly used model is the Auto-Regressive Moving Average (ARMA)-model, which is a rational transfer function from input samples to output samples. In other words, the ARMA-model is a system with transfer function  $H(z) = \frac{B(z)}{A(z)}$  where the coefficients in the polynomials are the parameters. Setting B(z) = 1 results in an Auto-Regressive (AR)model; setting A(z) = 1 results in a Moving Average (MA)-model.

It turns out that any ARMA-model can be represented by an AR-model, although it may require more parameters. The choice in practice is usually an AR-model, because it yields simpler equations and it can detect narrow peaks.

#### 2.5.3 Subspace estimation

Subspace methods use eigendecomposition or eigenanalysis of the correlation matrix to estimate frequency components. These methods are also known as high-resolution or superresolution methods, as they are best suited for spectra with sinusoidal components, especially when sines are buried in noise and the SNR is low. Examples of these methods are Multiple Signal Classification (MUSIC) and Estimation of Signal Parameters via Rotational Invariance Techniques (ESPRIT), which are described for example in [28].

```
Listing 2.1: Matlab-code alternative following (2.14) for estimating the spectrum. x and y are equal-length arrays of samples.
```

```
l = length(x);
ccf = xcorr(x,y,'biased'); % yields result of length (21-1)
CCF = fft(ccf, 2*1); % fft using 21 points: zero-pad ccf with one 0
angles = exp(-i*2*pi*[0:1-1]/1);
spectrum = CCF(1:2:length(CCF)).*angles/1;
```

Listing 2.2: Matlab-code alternative following (2.15) for estimating the spectrum. x and y are equal-length arrays of samples.

```
spectrum = fft(x).*conj(fft(y)) / length(x)^2;
```

#### 2.6 Spectral estimation

Since an SA is generally used for all kinds of applications, it cannot assume anything about spectra, making non-parametric estimation the natural choice. Harmsen [39] experimentally verified classic non-parametric estimation to outperform parametric estimation in case nothing is known about a spectrum.

In the time-discrete domain, spectra of signals are periodic in  $2\pi$  radians, and therefore any continuous range of  $2\pi$  radians suffices to describe the spectrum. In this thesis, frequencies in the time-discrete domain are normalized to  $-\frac{1}{2} \leq f < \frac{1}{2}$ , where the sampling frequency  $f_s$  is normalized to 1; in effect  $-\frac{f_s}{2} \leq f < \frac{f_s}{2}$ .

Although the definition of G[k] in eq. (2.4) implies a discrete function of frequency, it is possible to evaluate it at any frequency. The result is just a (convenient) form of interpolation. Using  $-\frac{1}{2} \leq f < \frac{1}{2}$ , f can be set to f = k/N to arrive at

$$G(f) = \sum_{n=0}^{N-1} x[n]e^{-j2\pi fn}$$
(2.13)

Taking the DFT of the biased correlation estimator yields an estimate for the crossspectrum:

$$C_{XY}(f) = \sum_{n=-(N-1)}^{N-1} c_{XY}[n] e^{-j2\pi f n}$$
(2.14)

Using eq. (2.7), this can also be expressed as

$$C_{XY}(f) = \frac{1}{N} \overline{X(f)} Y(f)$$
(2.15)

where X(f) and Y(f) are the DFTs of x[n] and y[n]. The proof follows directly from the convolution theorem [1] by using eq. (2.12). Equations (2.14) and (2.15) show two mathematically equivalent ways to estimate the cross-spectrum. With Y = X, this latter form is also known as the periodogram [28]. The equivalence of both methods is illustrated in listings 2.1 and 2.2.<sup>3</sup>

<sup>&</sup>lt;sup>3</sup>The reader may be confused by the fact that eq. (2.14) takes a (2N - 1)-point summation while X(f)and Y(f) are the result of N-point summations. The answer to this problem is that essentially any frequency may be calculated using the DFT, as this is merely a matter of interpolation (see eq. (2.13)). For computational efficiency usually only equally spaced frequencies are calculated. In fact, using the FFT and the Decimation-In-Frequency (DIF)-algorithm [25], the same number of calculations are required for the DFT for both methods, yielding the exact same results.



Figure 2.3: The triangular bias  $w_B[k]$  (Bartlett window) and its amplitude response  $|W_B(f)|$  for correlation of 8 and 64 samples, where the shown amplitude should be multiplied by the number of samples to obtain the true values.

Similar to  $c_{XY}[n]$ , the statistical properties of  $C_{XY}(f)$  can be derived. All calculations are completely analogous to the calculations for continuous-time correlation discussed in [32], and are elaborated in sections B.3 and B.4. The expectation is

$$E[C_{XY}(f)] = W_B(f) * \Gamma_{XY}(f)$$
(2.16)

where

$$W_B(f) * \Gamma_{XY}(f) \stackrel{\triangle}{=} \int_{-1/2}^{1/2} W_B(f) \Gamma_{XY}(f-g) \,\mathrm{d}g$$

denotes the linear convolution of  $W_B(f)$  and  $\Gamma_{XY}(f)$ , and  $W_B(f)$  is the DTFT of the triangular bias (also known as the Bartlett-window, hence the subscript B)

$$w_B[k] = 1 - \frac{|k|}{N}, \quad |k| \le N$$
$$W_B(f) = \left(\frac{\sin \pi f N}{\sin \pi f}\right)^2$$

The bias therefore is

$$B[C_{XY}(f)] = \Gamma_{XY}(f) - W_B(f) * \Gamma_{XY}(f)$$

and the variance (under the assumption that the spectrum is smooth, i.e. it has a bounded derivative) is

$$\operatorname{var}\left(C_{XY}(f)\right) = \Gamma_{XX}(f)\Gamma_{YY}(f)\left(\frac{\sin 2\pi fN}{N\sin 2\pi f}\right)^2 + \left|\Gamma_{XY}(f)\right|^2 \tag{2.17}$$

Note that  $W_B(f)$  needs to be a  $\delta$ -function to be unbiased for any  $\Gamma_{XY}(f)$ , but this requires infinite measurement time.  $C_{XY}(f)$  is an inconsistent estimator, because

$$\lim_{N \to \infty} \operatorname{var} \left( C_{XY}(f) \right) = \left| \Gamma_{XY}(f) \right|^2 \tag{2.18}$$

So, even though  $c_{XY}$  is a consistent estimator of the ccf, its DFT is not a consistent estimator of the cross-spectrum!

In other words, if x[n] and y[n] are realizations of stochastic processes,  $C_{XY}(f)$  does not converge in any statistical sense to a limiting value as N tends to infinity. This can



Figure 2.4: Smoothing of the correlation function allows the computation of fewer lags.

be intuitively explained by the fact that more samples give more information, but using all values of the correlation function estimate (or, equivalently, all samples in the DFT) gives equally more frequency point estimates. Therefore, the amount of information per frequency point remains constant. As a result, the variance of each of these points does not decrease.

#### 2.6.1 Improving the estimation

Because an SA should provide statistically meaningful results, the variance should be decreased. There are several methods to do this.

The first, known as the Bartlett-method, is to split the N samples into K sequences of M = N/K samples. Averaging the power spectral estimations as defined in eq. (2.15) for each of these K sequences reduces the variance of the final estimate by a factor 1/K, at the cost of a loss in frequency resolution by a factor K. The result is a consistent estimator.<sup>4</sup>

An extension to the Bartlett-method was proposed by Welch [41] to allow partial overlap of the segments and any windowing function. The bias and variance of the power spectrum estimation are a function of this overlap and the window used. Typically, selecting a window means a trade-off between bias, variance and frequency resolution of the spectral estimate. More overlap requires more computations, and stops giving significant improvements at some point, depending on the window used. Typical overlap values range from 25% to 75%. A very similar effect can be obtained by averaging adjacent frequency bins, which is equivalent to periodogram averaging with a rectangular spectral window applied to the data series, also known as the Daniell window [42].

The second method is to put a window on the correlation function estimate to mitigate the effects of the unreliable estimates for large lags. This method of smoothing the periodogram was proposed and analyzed by Blackman and Tukey, and is known as the Blackman-Tukey method. One small disadvantage of this method is that for certain windows the power spectrum estimate can have negative values, which is physically impossible. It turns out that these negative values are in practice so small that they can be considered zero [42].

The latter method seems very convenient for the purpose of crosscorrelation, as a smoothing window  $w_S$ , centered at k = 0 (or, if  $w_S$  has even length, centered at k = -1/2), reduces the number of lags for which the ccf estimate has to be calculated (see fig. 2.4). This smoothing window is the reason why the estimators  $c_{XY}$  and  $\hat{c}_{XY}$  can be treated in the same way for spectral estimation: one can transform  $c_{XY}$  into  $\hat{c}_{XY}$  by applying the Bartlett-window to  $c_{XY}$ . Applying  $w_S[k]$  to the correlation function results in a total win-

<sup>&</sup>lt;sup>4</sup>One might wonder why the spectrum  $|X(f)|^2$  is averaged, and not the frequency domain representation X(f). The reason is that in general the phase of the desired signal will be different in each measurement, so the signal is also canceled out. Only if one knows the (approximate) phase of the signal can this be done, such that the signal contributions in each measurement add up coherently. This property is exploited in a signal analyzer made by Rohde & Schwarz [40] to perform high-sensitivity phase noise measurements.



Figure 2.5: For large K, the effective window in AC and XC tends to a rectangular window when a rectangular smoothing window is used.

dow  $w_T[k] = w_B[k] \cdot w_S[k]$ , where samples are considered 0 in case one window is shorter than the other.

#### 2.7 Correlation in a spectrum analyzer

As eqs. (2.14) and (2.15) show there are (at least) two approaches to determine the (cross)-spectrum. These methods can be used for two identical signals, i.e. X = Y, or for two different signals, i.e.  $X \neq Y$ . This means there are effectively four different situations:

- Autocorrelation method (AC), which follows eq. (2.14) and X = Y.
- Crosscorrelation method (XC), which follows eq. (2.14) and  $X \neq Y$ .
- Spectrum Averaging method (SAVG), which follows eq. (2.15) and X = Y.
- Cross-spectrum Averaging method (XSA), which follows eq. (2.15) and  $X \neq Y$ .

Without any modifications with respect to improving the estimation, AC and SAVG are mathematically equivalent, and so are XC and XSA. Of course, XC and XSA are generalizations of AC and SAVG respectively, but it turns out that it is not so easy to generalize their properties.

As was discussed in section 2.6, the estimation procedure needs to be improved to obtain a lower variance. This is where AC and XC start to differ from SAVG and XSA. In AC and XC, one windows the correlation function, while in SAVG and XSA different (possibly windowed) spectral estimates are averaged. This difference is shown in fig. 2.5; for SAVG and XSA the effective window remains the window of fig. 2.5a, due to the biased estimation.

A system model for an SA using the two main approaches is schematically depicted in fig. 2.6. The system is modeled as a signal source S, split up into two paths (X and Y),



(a) AC and XC window the correlation function (Blackman-Tukey method)



(b) SAVG and XSA average multiple spectra (Bartlett method)

Figure 2.6: Two different spectral estimation methods to lower the variance. For AC and SAVG a(t) = b(t).

where each path contributes a certain amount of additive noise, which are named A and B respectively. The resulting signals X = S + A and Y = S + B are then crosscorrelated. A and B have equal variance (which in the case the noise is Gaussian, as assumed here, is equal to the noise power), because the two chains are copies of each other, and will therefore contribute a similar amount of noise. It is assumed that A, B and S are ergodic for the duration of the measurement, and all fully uncorrelated. In practice this will not be completely true, because noise sources such as power supply and substrate bounce will (at least partially) be correlated. The signals a(t), b(t) and s(t) are realizations of these stochastic processes. In the case of AC and SAVG, the two noise sources are now fully correlated, i.e. a(t) = b(t). To be able to distinguish between these noise sources, this source is now called C with realization c(t).

To allow a fair comparison, the total number of samples (per branch) are equal for all four methods. The number of samples taken in each branch to perform a spectrum estimate for AC and XC is denoted as N, such that the estimated correlation functions (before smoothing) have a length of 2N - 1. The number of samples taken to perform a spectrum estimate using SAVG and XSA is denoted by M. The smoothing window in AC and XC has a length equal to 2M - 1 and is centered around lag k = 0. The number of averages for SAVG and XSA is denoted by K, with K = N/M.

From the definition of crosscorrelation, it can be easily shown with X = S + A and Y = S + B that

$$\gamma_{XY}(\tau) = \gamma_{SS}(\tau) + \gamma_{SB}(\tau) + \gamma_{AS}(\tau) + \gamma_{AB}(\tau)$$

#### 2.7. CORRELATION IN A SPECTRUM ANALYZER

and hence for the spectrum, using linearity of the Fourier-transform family,

$$\Gamma_{XY}(f) = \Gamma_{SS}(f) + \Gamma_{SB}(f) + \Gamma_{AS}(f) + \Gamma_{AB}(f)$$

Because the signal and the noise are uncorrelated, for Y = X the result (with  $\Gamma_{AB} = \Gamma_{CC}$ )

$$\Gamma_{XX}(f) = \Gamma_{SS}(f) + \Gamma_{CC}(f) \tag{2.19}$$

is found, while for  $Y \neq X$  the noise sources A and B are also uncorrelated, yielding

$$\Gamma_{XY}(f) = \Gamma_{SS}(f) \tag{2.20}$$

which is exactly the desired spectrum.

SAVG is the method that is usually available in commercial SAs. XC and XSA allow determining the cross-spectrum, which is the main idea of this thesis. AC is included as a reference method. Deriving the properties of these methods enables a comparison between them. This should show whether crosscorrelation is a feasible method.

#### 2.7.1 Asymptotic properties

Important properties of the estimators are the expected value and the variance. The closer the expected value to the true value, and the lower the variance, the more accurate the final results. Here the results are given for the situation in which tapering windows are all equal to a rectangular window.

#### Spectrum averaging (SAVG)

Only the results are shown; for the derivation see section B.5.

$$E[C_{XX}(f)] \approx W_B(f) * \Gamma_{XX}(f) \approx \Gamma_{XX}(f)$$
(2.21)

where the final step is valid if M is large enough, i.e. if  $W_B(f)$  tends to a  $\delta$ -function.

$$\operatorname{var}\left(C_{XX}(f)\right) \approx \frac{1}{K} \left|\Gamma_{XX}(f)\right|^2 \tag{2.22}$$

#### Autocorrelation (AC)

The asymptotic expressions for spectral estimation using autocorrelation are given by Jenkins & Watts [32]. The expectation, under the assumption that the total number of samples is much larger than the number of lags used for the spectral estimation, such that the influence of the Bartlett-window becomes negligible (see fig. 2.5), is [32, p. 245, eq. (6.3.35)]:

$$E\left[\tilde{C}_{XX}(f)\right] \approx W_S(f) * \Gamma_{XX}(f)$$

where  $W_S(f)$  is the spectrum of the smoothing window. If M is large enough, i.e. when  $W_S(f)$  (in this case the frequency response of a rectangular time window), tends to a  $\delta$ -function, this simplifies to

$$E\left[\tilde{C}_{XX}(f)\right] \approx \Gamma_{XX}(f)$$
 (2.23)

The asymptotic variance is [32, p. 251, eq. (6.4.12)]:

$$\operatorname{var}\left(\tilde{C}_{XX}(f)\right) \approx \frac{I}{N} \Gamma_{XX}^2(f) \approx \frac{2}{K} \Gamma_{XX}^2(f)$$
(2.24)

where

$$I = \sum_{m} w_{S}^{2}[m] = \int_{-\frac{1}{2}}^{\frac{1}{2}} W_{S}^{2}(g) \,\mathrm{d}g$$
(2.25)

and I/N can be simplified to 2/K because, if M and K are large enough, K = N/M and  $I = 2M - 1 \approx 2M$ .

It can be observed that although the expectation of AC is very similar to the expectation of SAVG, the variance of AC is twice as large. This is caused by the different effective windows as shown in fig. 2.5. The effective window will have an influence on the frequency resolution as shown in fig. 2.2. Although this comparison may not be the fairest, it certainly shows that the dependency as a function of the number of samples is equal.

#### Cross-spectrum averaging (XSA)

Only the results are shown; for the derivation see section B.6. For XSA the estimator is denoted as  $|\tilde{A}_{XY}|$  for reasons discussed in section B.6.

$$E\left[\left|\tilde{A}_{XY}\right|\right] \approx \sqrt{\Gamma_{SS}^2 + \frac{\beta}{K}\left(\Gamma_{SS}\Gamma_{AA} + \Gamma_{SS}\Gamma_{BB} + \Gamma_{AA}\Gamma_{BB}\right)}$$
(2.26)

and

$$E\left[\left|\tilde{A}_{XY}\right|^{2}\right] = \frac{K+1}{K}\Gamma_{SS}^{2} + \frac{1}{K}\left(\Gamma_{SS}\Gamma_{AA} + \Gamma_{SS}\Gamma_{BB} + \Gamma_{AA}\Gamma_{BB}\right)$$
(2.27)

with

$$\beta = \frac{\pi}{4K} \left( \frac{\Xi \left( K + \frac{1}{2} \right)}{\Xi(K)} \right)^2 \left( 1 - \frac{\Gamma_{SS}^2}{E \left[ \left| \tilde{A}_{XY} \right|^2 \right]} \right) + \frac{1}{2} \frac{\Gamma_{SS}^2}{E \left[ \left| \tilde{A}_{XY} \right|^2 \right]}$$
(2.28)

where

$$\Xi(x) = \int_0^\infty e^{-t} t^{x-1} \,\mathrm{d} x$$

is the mathematical Gamma-function, but written as  $\Xi$  to avoid confusion with the spectra. The variance can then be calculated using the well-known formula

$$\operatorname{var}\left(\left|\tilde{A}_{XY}\right|^{2}\right) = E\left[\left|\tilde{A}_{XY}\right|^{2}\right] - E^{2}\left[\left|\tilde{A}_{XY}\right|\right]$$
(2.29)

 $\beta$  will always be between  $\frac{1}{2}$  (for the extreme  $\Gamma_{SS} \gg \Gamma_{AA}, \Gamma_{BB}$ ) and  $\frac{\pi}{4}$  (for the extreme  $\Gamma_{SS} \ll \Gamma_{AA}, \Gamma_{BB}$ ), so for back-of-the-envelope calculations one can use e.g.  $\beta = \frac{2}{3}$  or  $\sqrt{\beta} = \frac{4}{5}$ , whichever comes in handy.

#### Crosscorrelation (XC)

Exploiting the resemblance between SAVG and AC on one hand and XSA and XC on the other hand, one may be inclined to guess that the variance of XC will be related to the variance of XSA by a factor of 2. Indeed, simulation results (discussed later) indicate that replacing all K in eqs. (2.26)–(2.28) by K/2 yields a good approximation for the expectation and variance of XC. No attempt has been made to mathematically verify this substitution.

#### Simulation

To validate the formulas, the asymptotic approximations are compared to simulation results as shown in fig. 2.7. The simulations are performed up to K = 200 (but only shown for low K).

For all four methods the approximations and simulation results converge. Simulations (not shown) were also performed for much higher and much lower noise levels, which both show convergence. This indicates that the asymptotic approximations can be used to assess the performance of the four methods.

22





(c) Standard deviation for frequencies (d) Standard deviation for frequencies where  $\Gamma_{SS}(f) = 0$  where  $\Gamma_{SS}(f) = 1$ 

Figure 2.7: Comparison of simulation results (markers) and asymptotic approximations (solid lines) for expectation and standard deviation. Simulations were performed 500 times and averaged using bandpass-filtered white noise as signal with  $\Gamma_{SS}(f) = 1$  and a white noise floor with  $\Gamma_{NN}(f) = 4$ .

The derived trends are very important. For all methods the variance decreases linearly with the number of samples, which means that the standard deviation scales with the square root of this number. For SAVG and AC the expected value does not change with the number of samples, while for XSA and XC the expected value of the uncorrelated parts decreases with the square root of the number of samples. This means that using crosscorrelation, in order to lower the noise floor by 3 dB, a four times longer measurement time is needed.

#### 2.7.2 Signal detection

The SFDR was defined as the ratio between the largest and the smallest signal that can be detected at the same time. This smallest signal is usually set equal to the noise floor level [4, 19]. With that definition, the SFDR cannot be improved with AC or SAVG, but it can with XC and XSA. Intuitively, this is not a satisfying situation, because AC and SAVG reduce the variance of the noise, thereby increasing the ability to detect a signal. In the limiting case (with an infinite number of averages), the noise floor will be completely flat and *any* signal can theoretically be detected, no matter how far it is below the noise floor.

There are several practical issues possibly limiting this averaging, other than the limited time the input signal is present. These include the finite set of values that can be represented in the digital domain, measurement errors/uncertainty, a not-completely-white noise floor,



Figure 2.8: Graphical representation of observability

and instability of the receiver chain (all components in the analog domain) which changes the noise level in time. Whether any of these effects has a significant influence requires further research.<sup>5</sup> Nevertheless, to be able to make a fair comparison, a definition is needed which incorporates the variance of the noise to determine whether a signal is *observable*.

A general rule in radioastronomy is that the smallest signal that can be measured equals the standard deviation at the output of a filter in case no signal is present [43]. In that case, signals are discovered by comparing the output in time, but in the current situation it could just as easily be compared in frequency.

The smallest signal that can be observed will be defined as a signal with a mean value of the power inside the frequency bins that is at least equal to the average value plus the standard deviation of the noise:

**Definition 2.1** (Observability). Let X be a band-limited stochastic process and Y a white noise process. Let X(f) and Y(f) denote their respective spectra. Let Z = X + Y be the sum of X and Y, and C(f) the spectral estimator of Z. Let  $f_1$  denote a frequency for which  $X(f_1) = 0$  and let  $f_2$  denote a frequency for which  $X(f_2) > 0$ . If

$$E[C(f_1)] + \sqrt{\operatorname{var}(C(f_1))} < E[C(f_2)]$$
(2.30)

then X is observable at frequency  $f_2$ .

This definition is depicted in fig. 2.8. Note that this is not the same as the general definition of *sensitivity* of a receiver, which is the minimum input signal required to produce an output signal of a given SNR [44]. The reason for deviating from this definition is that the correlation process makes the sensitivity a function of the number of samples taken. This is equivalent to stating that the NF of the system goes down, because sensitivity and NF are directly related [3, 44]. This result allows a correlation SA to perform extremely sensitive measurements [45, 46].

#### 2.7.3 Simulation

To validate whether the asymptotic results obtained for the four aforementioned methods can be used to predict the measurement time needed to observe a signal, simulations were performed to determine whether a signal is observable or not. The signal consists of a bandpass filtered white noise source with PSD = 1, and the noise source(s) consist(s) of white noise with PSD = 10, which means that the signal is buried 10 dB below the noise floor. With  $M = 2^8$ , the minimum number of samples is  $N = M = 2^8$ . The results are shown in fig. 2.9.

 $<sup>^{5}</sup>$ The latter two effects, a not-completely-white noise floor and instability of the receiver chain, were observed first-hand in a linearity measurement of a Tayloe mixer using an Agilent SA, the results of which will be published on ISSCC2009.



Figure 2.9: Simulation results (lines) and expected values (circles) for observability for SAVG, XSA, AC and XC, with  $M = 2^8$ , a rectangular smoothing window and N = KM the total number of samples used.

Because the signal is a band-pass filtered signal, there is a noise floor on either side. The noise levels on both sides do not necessarily have to be the same in that particular realization. The signal is considered observable if the definition applies to both sides, not observable if the definition does not apply to either side, and half observable if the definition applies only to one side. With many simulations, the percentage of times at which the signal is observable is calculated and plotted.  $M = 2^8$  is chosen as a trade-off between simulation speed and effect of non-ideal Finite Impulse Response (FIR)-filtering (a perfect brickwall filter is not realizable). To reduce the influence of the sidebands of the bandpass signal in calculating the average value and variance in each section, it is made sure that frequencies near the transition are not taken into account.

Using the asymptotic approximations, it can be calculated when the signal becomes observable. In the stochastic simulation, this would be the point at which the signal is observable 50% of the time (note that all formulas deal with *ensemble averages*). Using definition 2.1, the equation

$$E[C(f_1)] + \sqrt{\operatorname{var}(C(f_1))} = E[C(f_2)]$$
(2.31)

needs to be solved for all four methods.

This will be worked out for XSA; only the results will be given for the others. Using eq. (2.26) and  $\Gamma_{AA} = \Gamma_{BB} = \Gamma_{CC} = 10$  and  $\Gamma_{SS} = 1$  one finds at a frequency  $f_1$  where only noise is present that

$$E\left[\left|\tilde{A}_{XY}\right|\right] \approx \sqrt{\frac{\beta_1}{K}\Gamma_{AA}\Gamma_{BB}} \approx \sqrt{\frac{25\pi}{K}}$$
(2.32)

because  $\beta_1$  (the  $\beta$ -factor is different for both frequencies, hence the subscript) tends to  $\pi/4$  for large K. From eq. (2.29) follows

$$\operatorname{var}\left(\left|\tilde{A}_{XY}\right|\right) \approx \frac{100}{K} - \frac{25\pi}{K} \tag{2.33}$$

At a frequency  $f_2$  where both signal and noise are present eq. (2.26) gives

$$E\left[\left|\tilde{A}_{XY}\right|\right] \approx \sqrt{\frac{K+1}{K} + \frac{80}{K}}$$
(2.34)

when taking  $\beta_2 = \frac{2}{3}$ . Substituting eqs. (2.32)–(2.34) into eq. (2.31) and solving for K yields  $K \approx 101$ .



Figure 2.10: Spectra obtained  $(M = 2^{12})$  for an input signal (black) using crosscorrelation (light gray) and spectrum averaging (dark gray) for different number of samples.

Doing similar calculations for the other methods yields  $K \approx 100$  for spectrum averaging,  $K \approx 200$  for autocorrelation,  $K \approx 101$  for cross-spectrum averaging and  $K \approx 202$  for crosscorrelation. From fig. 2.9 one finds  $K \approx 99$  for spectrum averaging,  $K \approx 198$  for autocorrelation,  $K \approx 105$  for cross-spectrum averaging and  $K \approx 206$  for crosscorrelation, all very close to the values predicted by the approximation.

If one takes a look at the spectra obtained using SAVG and XSA, shown in fig. 2.10, one cannot deny the fact that the spectra obtained using XSA more clearly show the signals, while SAVG requires a good eye to distinguish a signal level 10 dB below the noise floor (and hence only a 0.4 dB increase in the level displayed on the screen).

The reason for the difference is the fact that the signals are displayed on a dB-scale, while the calculations are on a linear scale. A practical issue with modern SAs is the relative amplitude accuracy, which is the accuracy of the difference between two spectral values of a measurement. The relative amplitude accuracy is typically in the order of 0.1 dB to 1 dB [3]. One might therefore require at least 1 dB difference between a signal and the surrounding noise. An adaptation to eq. (2.30) could then be

$$10^{\frac{1}{10}} \left( E\left[C(f_1)\right] + \sqrt{\operatorname{var}\left(C(f_1)\right)} \right) < E\left[C(f_2)\right]$$

As a result of this adaptation, a signal more than 5.9 dB below the noise floor can never be observable using AC or XSA. Provided enough noise is uncorrelated, XC and XSA can enable the detection of those signals, which is in accordance with [47].

#### 2.8. CONCLUSIONS

#### 2.8 Conclusions

Correlation is a mathematical technique to relate a signal with itself or another signal. This can be done in the analog or digital domain, but, because of noise requirements and ease of implementation, only the digital approach has been considered.

Several categories of spectral estimation techniques exist, but a classical approach is chosen as it does not require any prior knowledge of the signal to be observed. A spectrum can be obtained by first correlating and then taking a DFT, or by taking a DFT directly on the obtained samples and then squaring the result. The two are mathematically equivalent in their basic form. In case one uses techniques such as windowing and averaging, they are no longer mathematically equivalent, but still very similar in their properties.

By splitting the signal to be measured into two equivalent paths, the noise introduced by each path will be highly uncorrelated. Using crosscorrelation (in the form of XC or XSA) the effective noise level can be reduced at the cost of measurement time. Most commercially available SAs offer SAVG, which only smoothens the noise floor. Although all methods have the effect of allowing smaller signals to be detected, AC and SAVG hit a hard stop because of limited relative amplitude accuracy. Crosscorrelation does not suffer from this problem and therefore allows much smaller signals to be detected, which can be a significant advantage.

To assess the improvement using crosscorrelation, asymptotic expressions for the expectation and variance of all four spectral estimation methods are obtained. Fortunately these asymptotic approximations seem to converge rather fast, such that there is no significant difference in a situation where signals are buried in noise, which is the most important reason to use crosscorrelation. Simulations confirm that the asymptotic expressions accurately predict the number of samples required to observe a signal.

An important conclusion is that the noise floor using XC or XSA goes down with the square root of the number of samples. To lower the noise floor by 3 dB, the measurement time goes up by a factor of 4. In short, crosscorrelation works, but it is still important to have a device with a relatively low noise floor.

#### 2.9 Recommendations

The asymptotic approximations provide good insight into the number of averages or the number of samples that need to be taken to observe a signal given the current definition. In the simulation results of fig. 2.7, it looks as if the asymptotic approximations provide a kind of upper bound on expectation and variance. Although this could be a coincidence for this particular situation, if it turns out to be true for all situations, the approximations become much more useful.

From the simulation results in fig. 2.9 the approximations can be seen to 'fail' for AC and XC for a low number of samples. This is probably due to the fact that it takes some time before the effective window really approximates a rectangular window. It may be possible to include the effect of the partial Bartlett-window (see fig. 2.5) such that the approximations also hold for a lower number of samples.

Biasing effects and spectral leakage were all disregarded and made insignificant in the simulation by taking enough points for the FFT. In practical situations one might want to take a 16-point FFT, and in that case these effects come into play. An approximation that works for all cases would be much more useful.

Asymptotic approximations have been established for the situation in which both the noise sources and the signal to be measured are stochastic processes. If the common signal is deterministic, the situation will be different. If S is deterministic, the factor  $\Gamma_{SS}^2$  will not introduce a variance, but because A and B are noise sources, the terms  $\Gamma_{SS}\Gamma_{AA}$ ,  $\Gamma_{SS}\Gamma_{BB}$  (or  $\Gamma_{SS}\Gamma_{CC}$  for SAVG and AC) still do. This could make an important difference for the amplitude accuracy in measuring sinusoids. It would be useful to find out if the derived formulas can be easily adapted to this situation, and if so, what that adaptation would be.
Chapter 3

# Quantization

A necessary operation in a system that processes a signal in both the analog and the digital domain is AD-conversion. The continuous range of values of an analog signal needs to be converted to a finite set of discrete levels. This process is called *quantization*.

In this chapter the effects of quantization of a sinusoid on the SFDR will be discussed. This is important to know, because the SA-design as a whole needs an SFDR of 70 dB.

Sampling introduces aliasing, i.e. frequencies with  $f > f_s/2$  will alias back to a frequency  $0 \le f < f_s/2$ . It is assumed that anti-alias filters remove all such components to a negligible level, such that the only aliasing effects that remain are caused by signal distortion due to quantization.

## 3.1 Introduction

Digital processing of an analog signal requires the analog signal to be converted from the analog to the digital domain. This conversion process is called *analog-to-digital conversion* and is performed by an ADC. AD-conversion consists of two steps: Sample & Hold (SH) followed by quantization, which is depicted in fig. 3.1.

The performance of ADCs is mainly limited by noise due to the SH-circuit, and signal distortion due to quantization [48]. A higher performance (less noise, higher resolution, higher linearity) generally requires a higher power consumption [49]. Power consumption is decreasing over the years, but sampling rate and resolution improve only slowly [49], although this is disputed by others [48].

Quantization limits the output to a finite set of discrete values, which in general will not exactly represent the input signal. In most cases the quantization levels are at uniform distance, known as the Least Significant Bit (LSB).

The quantization error is the difference between the output and the input signal of the quantizer. It is usually assumed that the error is uniformly distributed between  $-\frac{1}{2}$ LSB and  $\frac{1}{2}$ LSB with an equal amount of power for each frequency, i.e. it is modeled as white noise and independent of (and hence uncorrelated from) the input signal. The total power  $\epsilon$  of this quantization noise can then be calculated to be

$$\epsilon = \frac{1}{\text{LSB}} \int_{-\frac{1}{2}\text{LSB}}^{\frac{1}{2}\text{LSB}} x^2 \,\mathrm{d}x = \frac{\text{LSB}^2}{12} \tag{3.1}$$

With the above assumptions and a full-scale sinusoid as input signal, one can derive the SNR at the output of a b-bit quantizer. A sinusoid with amplitude A has power  $\frac{1}{2}A^2$ , so

Large parts of this chapter have been submitted to **IEEE Transactions on Circuits and Systems II: Express Briefs** as *"Spurious-Free Dynamic Range of a Uniform Quantizer"* by M.S. Oude Alink, A.B.J. Kokkeler, E.A.M. Klumperink, K.C. Rovers, G.J.M. Smit and B. Nauta.



Figure 3.1: Graphical representation of analog-to-digital conversion. Sampling and quantization can theoretically be performed in arbitrary order, but usually the sampled signal is quantized.

when it is full-scale the power is  $\frac{1}{2}(\text{LSB} \cdot 2^{b-1})^2$ . The SNR then is

$$SNR = \frac{\frac{1}{2} \left( LSB \cdot 2^{b-1} \right)^2}{\frac{LSB^2}{12}} = \frac{3}{2} 2^{2b}$$
(3.2)

which expressed in dB becomes

SNR = 
$$10 \log \frac{3}{2} 2^{2b} = 6.02b + 1.76 \text{ [dB]}$$
 (3.3)

In reality the quantization error will not be uniformly distributed and can be correlated with the input signal. If the input signal is a noise source with known probability density function (pdf), the effects of quantization can be removed by correcting the resulting acf. A famous example is the correction of 1-bit quantization of a Gaussian noise source, known as the Van Vleck-correction [50]. Note that corrections fail if the noise has a different pdf or if it is not noise, and therefore this correction cannot be used in a general purpose SA.

If the input signal is a deterministic signal, uniform quantization gives rise to distortion, resulting in a spectrum different from the input signal plus an additive white noise floor. The total power of the distortion components is still relatively well approximated by eq. (3.3), especially for a larger number of quantization levels.

For a sinewave input without noise, uniform quantization results in pure harmonic distortion. This has been analyzed by Blachman [51], resulting in formulas that involve infinite and slowly converging summations of Bessel functions. The distortion power is not distributed evenly over the harmonics as the reader will see later. Because the quantized signal will be sampled in an ADC, aliasing will cause all distortion components to fall in the 0 to  $f_s/2$  frequency region, and therefore all distortion components need to be considered. The power of the distortion peaks decreases when the resolution of the quantizer is increased, but the resolution of ADCs is limited by the required sampling rate and the maximum allowable power consumption [48]. It would be useful to have simple design equations to allow for exploration of the design space and optimize between resolution, SNR and SFDR.

Although not directly of concern for SAs, it is useful to note that everything in this chapter applies equally well to zero-order hold DACs. The zero-order hold DAC is an important type of DAC. It retains the sample value until the next sample, so for uniformly quantized digital signals the result is a uniformly quantized analog signal, and the same situation is obtained as after quantizing an analog signal.

As was discussed in chapter 2, crosscorrelation makes it possible to lower the noise floor when the noise is uncorrelated. With each branch having its own ADC, this is a valid assumption for the thermal noise. In case the ADCs are triggered by separate clocks, the effects of clock jitter may also be mitigated by the correlation process. However, the distortion introduced by quantization will be partially correlated as the signal is present in both branches. As a result, the correlation process will not lower them, and the SFDR will be limited by this distortion. If an SFDR of 70 dB is required, the distortion components caused by quantization need to be more than 70 dB below the desired signal component. It is important to know the lowest amount of bits required, as this allows a higher sampling rate [48], lower power consumption [49] and simpler digital hardware. The downside of using fewer bits is the increase in quantization noise, which requires a longer correlation time if the quantization noise is not negligible compared to the noise introduced by the analog part of the system.

## 3.2 Quantization of a sinusoid

Multilevel quantization of a sinusoid without noise has been investigated mathematically by Blachman [51]. It was shown that for a midriser quantizer (i.e. a quantizer with a threshold exactly at 0) only odd-order harmonics are produced due to the odd-symmetric nature of the quantization staircase. To simplify calculations, all amplitudes in this chapter are expressed in LSB, and one LSB is normalized to 1. The resulting output signal then is equal to [51]

$$A_p = \delta_{p,1}A + \sum_{m=1}^{\infty} \frac{2}{m\pi} J_p(2m\pi A)$$
(3.4)

where  $A_p$  is the output amplitude of the *p*-th harmonic,  $\delta_{i,j}$  is the Kronecker delta function, A is the input amplitude and  $J_p$  is the *p*-th order Bessel function of the first kind.

Using the quantization staircase q(x) as shown in fig. 3.2, eq. (3.4) can be generalized to uniform quantizers by writing it as a linear transfer plus the quantization error. This quantization error is periodic with a period of 1 LSB. Hence, q(x) can be written as the sum of x and the Fourier series of the quantization error (note that for notational convenience a minus-sign is used for the cosine part).

$$q(x) = x + \sum_{m=1}^{\infty} a_m \sin(2\pi m x) - \sum_{m=1}^{\infty} b_m \cos(2\pi m x)$$
(3.5)

where the coefficients can be found by straightforward calculation:

$$a_m = \frac{2}{T} \int_{\langle T \rangle} (q(x) - x) \sin \frac{2\pi mx}{T} dx$$
  
=  $2 \int_{\Delta}^{\Delta+1} \left(\frac{1}{2} + \Delta - x\right) \sin(2\pi mx) dx = \frac{2\cos^2(\Delta\pi m) - 1}{\pi m}$   
 $b_m = -\frac{2}{T} \int_{\langle T \rangle} (q(x) - x) \cos \frac{2\pi mx}{T} dx$   
=  $-2 \int_{\Delta}^{\Delta+1} \left(\frac{1}{2} + \Delta - x\right) \cos(2\pi mx) dx = \frac{\sin 2\Delta\pi m}{\pi m}$ 

where  $\Delta$  is the offset in LSB.

Using the same method as Blachman [51] one finds

$$A_{p} = \begin{cases} \sum_{m=1}^{\infty} b_{m} J_{0}(2\pi mA) & p = 0\\ 2 \sum_{m=1}^{\infty} a_{m} J_{p}(2\pi mA) + \delta_{p,1} A & p \text{ odd} \\ 2 \sum_{m=1}^{\infty} b_{m} J_{p}(2\pi mA) & p \text{ even} \end{cases}$$
(3.6)

which reduces to eq. (3.4) for  $\Delta = 0$ .

Because A is expressed in LSB, the number of quantization levels n directly depends on the amplitude A of the sinusoid. Thus n does not need to be a power of two. A real-life



Figure 3.2: The quantization staircase of uniform quantizer can be decomposed into a straight line and a repetitive quantization error. The black lines represent  $\Delta = 0$ , corresponding to a midriser quantizer.



Figure 3.3: Partial spectrum of a full-scale sinusoid after 8-bit quantization.

situation where this is the case is a sinusoid that does not cover the whole input range of the quantizer.

Figure 3.3 shows the spectrum of a full-scale sinusoid quantized with 8 bits (A = 128), obtained by simulation and by numerical evaluation (both in Matlab) of eq. (3.4). For the harmonic p with the highest power, p equals 795, which is close to  $256\pi$  as deduced by Blachman [51], where it was shown that the highest spurious harmonic is around  $p \approx 2\pi A$ . Numerical evaluation (not shown here) indicates that the approximation of the strongest harmonic being located roughly at  $2\pi A$  is only valid for at least 20 quantization levels. In other cases the third harmonic is the strongest.

Pan & Abidi [52] simulated the effect of quantizing a sinusoid using a midriser quantizer. They constructed two linear fits for the power of the most powerful harmonic as a function of the number of bits, both with a slope of 9 dB/bit, but with different offsets. Although these fits were intuitively explained, it can be seen from their simulation results [52, fig. 3] that the true slope is somewhat less than 9 dB/bit.

The SFDR was determined by numerical evaluation for a full-scale sinusoid with  $\Delta = 0$  for all even n (to keep symmetry around zero) up to 13 bits, as shown in fig. 3.4a. A linear fit (also shown in fig. 3.4a) of these points results in

$$SFDR_{\sigma_N=0} = 8.07b + 3.29 \ [dB] \tag{3.7}$$

where  $b = \log_2 n$ . The subscript  $\sigma_N = 0$  is added to denote absence of noise. Because n is not necessarily a power of 2, b is not necessarily an integer. This approximation has a mean absolute error of 0.25 dB and a standard deviation of 0.31 dB, with a maximum error of 1.56 dB occurring for 58 quantization levels. Numerical evaluation showed this



(a) SFDR for a full-scale sinusoid as a function of the number of quantization levels for a midriser quantizer (points) and linear fit (line).



Figure 3.4: The SFDR in dB after quantizing a full-scale sinusoid with b bits can be approximated by a straight line.

approximation to hold up to at least 25 bits.

There is one ambiguity in the definition of the SFDR that needs to be clarified. In the case of only a few quantization levels, the first harmonic (or fundamental component) is lower in amplitude at the output than at the input. For example, in case of 1-bit quantization of a full-scale sinusoid, the power of the output fundamental is 3.8 dB lower than the input power. One could define the SFDR with respect to the input amplitude (which is the usual definition) or with respect to the output amplitude. The difference is shown in fig. 3.5. It can be seen that for 4 bits or more the difference is negligible. The linear fit of eq. (3.7) uses the SFDR with respect to the output fundamental, which seems to be in accordance with the choice made by Pan & Abidi [52].

For arbitrary  $\Delta$ , both even-order and odd-order harmonics are present, which means the distortion power is distributed over more distortion components. One would expect the peaks to decrease by roughly 3 dB if even-order and odd-order harmonics are equally strong. Setting  $a_m$  in eq. (3.6) equal to  $b_m$  for m = 1 (for reasons to be discussed in section 3.3):

$$\frac{2\cos^2\left(\Delta\pi\right) - 1}{\pi} = \frac{\sin 2\Delta\pi}{\pi} \tag{3.8}$$

yields, using the identities [53]

$$\cos\frac{\pi}{8} = \frac{1}{2}\sqrt{2+\sqrt{2}} \qquad \qquad \cos\frac{5\pi}{8} = \frac{1}{2}\sqrt{2-\sqrt{2}} \tag{3.9}$$



Figure 3.5: Difference in dB between SFDR related to the input power and related to the fundamental output power.



Figure 3.6: Zoom-in on spectrum for 8-bit quantization of a sinusoid with near-full-scale amplitude. The 'holes' in the spectra contain values below the range shown.

the solutions  $\Delta = \frac{1}{8}$  and  $\Delta = \frac{5}{8}$ . Numerical evaluation then indeed shows that the SFDR increases by roughly 3 dB as compared to eq. (3.7).

In practice, the amplitude will never be exactly full-scale. Numerical evaluation shows that the SFDR changes randomly with the same magnitude as shown in fig. 3.4 in the case the amplitude is somewhere in the range between full-scale and full-scale minus 1 LSB. An example is shown in fig. 3.6. This suggests that approximations to the theoretical value may be off by 1 dB or 2 dB without compromising practical relevance.

## 3.3 Mathematical derivation of the trend

In this section the trend observed in the previous section will be mathematically derived. It will turn out that it enables generalization to multitone inputs and the quantization of a sinusoid in the presence of noise.

Blachman [51] showed that the harmonics for  $p \ll 2\pi A$  decrease by 3.01 dB per bit. In his case, the LSB remains 1. In practice the maximum amplitude is often given and one would like to see what happens if the resolution is increased. An extra bit means that the LSB is divided by two. This corresponds to another 6.02 dB/bit decrease, so the net result is a decrease of 9.03 dB/bit. This was also the conclusion drawn by Pan & Abidi [52].

Here, however, it is found that the harmonics around  $p \approx 2\pi A$  decrease by only 2 dB/bit (leaving the LSB at 1). This will be derived from the exact analytic formula as given in eq. (3.6) by using approximations to the Bessel function and showing that only the first term in the summation contributes to the trend.

As suggested in [51], the Bessel functions can be approximated by Airy functions in the



Figure 3.7: Some factors as a function of the number of quantization levels.



Figure 3.8: Numerical evaluation of  $\hat{p}^{\frac{2}{3}}\zeta(\pi n/\hat{p})$  for a midriser quantizer with *n* ranging from 20 to 1024.

region where  $p \approx 2\pi A$  [54, p366, eq. (9.3.6)]:

$$J_p(pz) \approx \left(\frac{4\zeta(z)}{1-z^2}\right)^{\frac{1}{4}} \frac{\operatorname{Ai}\left(p^{\frac{2}{3}}\zeta(z)\right)}{p^{\frac{1}{3}}}$$
(3.10)

where

$$\zeta(z) = -\left(\frac{3}{2}\sqrt{z^2 - 1} - \frac{3}{2}\arccos\frac{1}{z}\right)^{\frac{2}{3}}$$
(3.11)

With the strongest harmonic  $\hat{p}$  located at  $\hat{p} \approx 2\pi A$ , eq. (3.4) in combination with eq. (3.10) yields

$$A_{\hat{p}} \approx 2 \sum_{m=1}^{\infty} c_m \left(\frac{4\zeta(z)}{1-z^2}\right)^{\frac{1}{4}} \frac{\operatorname{Ai}\left(\hat{p}^{\frac{2}{3}}\zeta(z)\right)}{\hat{p}^{\frac{1}{3}}}$$
(3.12)

where  $c_m = a_m$  if  $\hat{p}$  is odd,  $c_m = b_m$  if  $\hat{p}$  is even, and  $z = 2\pi m A/\hat{p}$ .

The cases m = 1 and m > 1 of eq. (3.12) are considered separately. It will turn out that only the m = 1 term in the summation contributes to the trend, while the other terms merely act as 'random' deviations from this trend.

Apart from the bell-shaped curve at the peaks, the quantization error of a quantized sine-wave resembles a sawtooth waveform with a modulated period, where the maximum frequency (at the zero-crossings) equals  $2\pi A$  times the frequency of the sine [52]. From numerical analysis it is seen that  $\hat{p}$  is always slightly smaller than  $2\pi A$ , but tends to get closer to  $2\pi A$  for larger A. For m = 1 one numerically finds  $z \in (1; 1.1)$  if  $n \ge 20$ , see fig. 3.7a. In this region the factor  $4\zeta(z)/(1-z^2)$  is virtually constant, see fig. 3.7b.

The parameter  $\hat{p}^{\frac{2}{3}}\zeta(2\pi A/\hat{p})$  of the Airy-function was numerically evaluated for a midriser quantizer quantizing a full-scale sinusoid for n = 20 to n = 1024 and is shown in fig. 3.8. Obviously this parameter stays roughly -1, indicating that  $\operatorname{Ai}(\hat{p}^{\frac{2}{3}}\zeta(2\pi A/\hat{p}))$  can also be considered a constant.



Figure 3.9: The Airy-function  $\operatorname{Ai}(-x)$  (solid line) and the approximation of eq. (3.14) (dashed line).

Removing constants, one obtains for the first term in the summation of eq. (3.12)

$$A_{\hat{p}}(m=1) \propto \frac{1}{\hat{p}^{\frac{1}{3}}}$$
 (3.13)

which corresponds to a  $\hat{p}^{-\frac{2}{3}}$  dependency in the power spectrum, equivalent to a decrease of 2.01 dB/bit.

For  $m \ge 2$ , the Airy-function can be approximated using [54, p449, eq. (10.4.83)]

$$\operatorname{Ai}(-x) \approx \frac{\sin\left(\frac{2}{3}x^{\frac{3}{2}} + \frac{\pi}{4}\right)}{\sqrt{\pi}x^{\frac{1}{4}}}$$
(3.14)

for  $x \gg 1$ . Figure 3.9 shows that this approximation is quite close for x > 1.2 (error less than 0.013).

Plugging this and the approximation  $z = 2\pi m A/\hat{p} \approx m$  into eq. (3.10) and simplifying one obtains

$$J_p(pz) \approx J_p(pm) \approx \left(\frac{4}{m^2 - 1}\right)^{\frac{1}{4}} \frac{\sin\left(-\frac{2}{3}p\zeta(m)^{\frac{3}{2}} + \frac{\pi}{4}\right)}{\sqrt{\pi}p^{\frac{1}{2}}}$$
(3.15)

There doesn't seem to be a relation between the number of quantization levels and the phase of the sinusoid in eq. (3.15). Hence, the sine-term can be considered as a random variable assuming values between -1 and 1 with an expectation of 0.

Concluding, only the first term in the summation of eq. (3.12) is important for the overall trend, while the other terms provide more or less random deviations. This randomness explains the erratic behaviour around the trend of the SFDR in fig. 3.4. Combining the 2.01 dB/bit for the m = 1 term and the 6.02 dB/bit from halving the amplitude of the quantization error, a trend of 8.03 dB/bit increase in SFDR is expected, which is very close to the 8.07 dB/bit obtained from numerical evaluation.

## 3.4 Quantization of a sinusoid with noise

Adding noise decorrelates the input signal and the quantization error [55]. The effect only depends on the univariate pdf of the noise, and not on its spectrum [51]. The resulting frequency response is the product of the frequency response of the quantization error as given in eq. (3.4) and the Fourier transform of the pdf of the noise [51, 55].

A general formula of the output spectrum of a crosscorrelator using quantization was derived by Kokkeler & Gunst [56] for arbitrary signals with arbitrary was noise sources. This model can be used to determine the output spectrum of a single quantizer, because, according to the Wiener-Khinchin theorem, the Fourier transform of the acf of a signal is equal to the spectrum of this signal.



Figure 3.10: Contour plot of the error made in dB by the approximation given in eq. (3.17).

Because thermal noise has a Gaussian pdf and is often the most important noise contribution, the effect of this noise on the SFDR will be studied in more detail. The resulting spectrum is found to be [51, 55, 56]:

$$A_p = \delta_1 A + \sum_{m=1}^{\infty} c_m J_p (2m\pi A) e^{-2\pi^2 \sigma_N^2 m^2}$$
(3.16)

where  $\sigma_N$  is the standard deviation of the noise in LSB.

Using the previously derived result that only m = 1 contributes to the trend of the SFDR as a function of the number of quantization levels, it can be observed that the increase in SFDR in dB depends quadratically on the noise level in LSB. Given as an equation:

$$SFDR_{\sigma_N} = SFDR_{\sigma_N=0} + 20\log_{10}e^{-2\pi^2\sigma_N^2} = 8.07b + 3.29 + 171.5\sigma_N^2 \text{ [dB]}$$
(3.17)

The result is that adding more noise increases the SFDR, which may look counter-intuitive. This is simply a matter of definition of the SFDR for a quantizer; only spurious peaks are considered. The SFDR of an SA does take the noise level into account, but this is only possible because the (effective) noise bandwidth is known.

Figure 3.10 shows a contour-plot of the error in dB between the approximated and calculated SFDR as a function of the number of quantization levels and the amount of noise. A negative error means the approximation underestimates the SFDR given by theory. Clearly the approximation is quite close over the whole range of values shown. The difference of less than 2 dB in virtually all cases bears no practical relevance as discussed in section 3.2.

Because the instantaneous amplitude of the noise can assume any value, practical quantizers will sometimes clip, resulting in (additional) distortion of the spectrum. Gaussian noise has an amplitude of less than  $3\sigma$  in 99.87% of the time. Simulations show that in case the amplitude is full-scale minus  $3\sigma$ , clipping effects have no visible influence on the spectrum.

## 3.5 Multitone quantization

Using the derivation of the 8 dB/bit trend for quantization of a single sinusoid, it turns out to be rather straightforward to generalize it to an arbitrary number of sinusoids. Suppose N sinusoids of frequency  $f_i$  and amplitude  $A_i$  are present at the input, where i ranges from 1 to N. The output will contain peaks at frequencies  $\sum_i p_i f_i$ , with  $\sum_i p_i > 0$  (note that any individual  $p_i$  can be negative), the amplitude of which is denoted by  $A_{p_1,\ldots,p_N}$ . This amplitude can be derived using the identities [51]

$$_{2}^{jz\sin(\theta)} = \sum_{p=-\infty}^{\infty} J_{p}(z)e^{jp\theta} \qquad \qquad J_{-p}(z) = (-1)^{p}J_{p}(z)$$
(3.18)

resulting in

$$A_{p_1, p_2, \dots, p_N} = \sum_{i=1}^N \left( \delta_{p_i, 1} A_i \prod_{j \neq i} \delta_{p_j, 0} \right) + \sum_{m=1}^\infty \left( c_m \prod_{i=1}^N J_{p_i}(2\pi m A_i) \right)$$
(3.19)

Note that if (some of) the input frequencies are commensurate, i.e.  $\exists k, l_j \in \mathbb{Z}$  such that  $kf_i = \sum_{j \neq i} l_j f_j$ , multiple amplitudes  $A_{p_1,\ldots,p_N}$  will occupy the same frequency, and need to be added while taking into account their respective phase relations to get the total amplitude at that specific frequency. This gives a much more complicated expression for the general case and is not discussed here.

In the derivation of the trend in SFDR for a single sinusoid, a  $p^{-\frac{1}{3}}$  amplitude dependence was found in the case the LSB was kept equal to 1. Since the summation now contains the product of N of these Bessel-functions, one finds a  $p^{-\frac{N}{3}}$  amplitude dependence when the frequencies are non-commensurate. This results in a 6.02+2.01N dB/bit increase in SFDR.

Simulations with two non-commensurate tones confirmed a trend of roughly 10 dB/bit. For more tones, simulation results become unreliable because the limited number of bins in an FFT cause multiple non-negligible intermodulation products to fall into the same frequency bin.

#### 3.6 Example

The approximation given in eq. (3.17) allows a quick evaluation of the trade-off between resolution of the ADC on one side and SNR and SFDR on the other. If the signal applied to a quantizer contains a sinusoid plus noise with a given SNR,  $\sigma_N$  can be readily evaluated by making use of the fact that a sinusoid with amplitude A has power  $A^2/2$ :

$$\sigma_N = \frac{n}{2\sqrt{2}} 10^{-\frac{\text{SNR}}{20}} \text{ [LSB]}$$
(3.20)

Consider the situation where the input of the quantizer has an SNR of 70 dB, and the demands are that the SNR at the output must be at least 65 dB, while the SFDR should be at least 90 dB. It is then found that the SNR of the quantizer needs to be at least 66.7 dB, which using eq. (3.3) yields  $n \ge 1753$ . For an SFDR of 90 dB, eqs. (3.20) and (3.17) are used to find  $n \ge 1273$ . In this case, the SNR-requirement is the limiting factor.

Consider the same situation with the SNR set to 60 dB with the demand that it must be at least 55 dB at the output. Using eq. (3.3) one finds  $n \ge 554$ . In this situation the requirement on the SFDR yields  $n \ge 698$ , which makes it the limiting factor. Numerically evaluating n = 698 using the exact formulas, the SFDR is found to be 91 dB.

This example clearly demonstrates the accuracy and ease of use of the approximation.

#### 3.7 Practical considerations

All the above calculations assume ideal ADCs, where the center of the quantization steps are all on a straight line. If they are not, such that Differential Non-Linearity (DNL) and Integral Non-Linearity (INL) are not equal to zero, there will be non-linear distortion, which can be "significant" [51], although no attempt was made to analyze this.

In a practical implementation, an FFT of the correlation function will generally have a limited number of points, usually in the order of  $2^4$  to  $2^{12}$ . This means that many

#### 3.8. CONCLUSIONS

harmonics may fall into the same frequency bin, thereby adding their powers. As a result, the SFDR is decreased. The combination of harmonics falling into the same frequency bin is fully dependent on the fraction  $f_0/f_s$ , where  $f_0$  is the original sinusoidal frequency and  $f_s$  the sampling frequency, which makes a general expression very hard to derive. Therefore it would be better to require the highest harmonic to be several dB lower than strictly necessary given the desired SFDR.

Except for the harmonics, quantization noise in both branches is uncorrelated, and can be lowered by the correlation process. Since lowering the noise floor by 3 dB already requires a 4 times longer correlation time, it would be advantageous if the noise floor is not increased significantly by the quantization noise. For evaluating the effect eq. (3.3) can be used. With an analog SNR of SNR<sub>analog</sub>, quantization noise can be considered insignificant if it increases the noise by less than x dB, where x is a design parameter. This means that

$$SNR_{ADC} \ge SNR_{analog} - 10 \log_{10} \left( 10^{\frac{2}{10}} - 1 \right) [dB]$$
 (3.21)

As an example, taking x = 0.5 dB and SNR<sub>analog</sub> = 40 dB, it follows SNR<sub>ADC</sub>  $\geq$  49 dB, which requires 8 bits. This extra 0.5 dB in noise power requires 25% longer integration time to obtain the same noise level as before quantization. Taking x = 1.5 dB requires 7 bits, and 100% longer integration time, while taking x = 0.1 dB requires 9 bits, and 4.7% longer integration time.

## 3.8 Conclusions

Quantization of a sinusoid introduces harmonic distortion, of which the most powerful component goes down by about 8 dB/bit. This trend was also mathematically verified by showing that only the first term in the analytical formula contributes to the trend. These results were then used to derive the effects of additive Gaussian noise and multitone inputs on SFDR. Both noise and multitone inputs decrease the correlation between input signal and quantization error, and hence increase SFDR. It can be regarded as a form of dithering.

The SFDR increases linearly with the variance of the Gaussian noise. If the standard deviation of the noise is equal to 1 LSB, the SFDR is increased by 171.5 dB. When a quantizer is excited by N sinusoids of non-commensurate frequencies, the SFDR increases by 6 + 2N dB/bit.

These results can be applied in designing systems without having to use the exact formulas, which can save a lot of time. It relates the SNR, the SFDR and the number of quantization levels; knowing two of them allows easy calculation of the third.

## 3.9 Recommendations

Pan & Abidi [52] showed through simulation that the SFDR is influenced by INL in a notso-straightforward way. In virtually all of their formulas they use their trend of 9 dB/bit, while here a trend of 8 dB/bit is found. It would be interesting to see if the effects of INL and DNL can be incorporated into the approximations derived in this chapter, for example by following the approach of [57]. This would make the approximations in this chapter even more useful in modern circuits and systems design.

## Chapter 4

# Design of the spectrum analyzer

The mathematical analysis of crosscorrelation in chapter 2 shows that crosscorrelation is a promising way to lower the noise level. In this chapter an SA will be designed that puts this effect to use to increase the SFDR. With the results of chapter 3, the minimal resolution of the ADC to obtain a desired SFDR can be calculated. Key properties of the system are linearity, NF, power consumption, cost and chip area, but as always trade-offs between them are unavoidable.

## 4.1 System design

Because crosscorrelation lowers the noise floor, a system with a very high linearity is desired. Lowering the noise floor by 3 dB using crosscorrelation requires a fourfold increase in measurement time, so the NF should not become too large.

Traditional receiver architectures start with an LNA at the input to limit the total NF of the system. For an SA, a wideband LNA would be required, but their Third Order Inputreferred Intermodulation Intercept Point (IP3) is typically limited to a few dBm [58] (the most linear wideband LNA found has an IP3 of +9 dBm [59]); much lower than desired. Moreover, it requires an even more linear mixer because its effective IP3 is decreased by the gain of the LNA. Therefore an architecture is envisioned without a wideband LNA, which increases linearity at the cost of noise.

On-chip filters cannot achieve very high specifications due to low-Q components and inductors that occupy a large area. External filters are expensive and may require extra pins on the package. An architecture that uses as little external filters as possible is preferred. Since their use can probably not be completely avoided, they are preferred directly after the antenna, where the pins on the package and impedance matching are required anyway. Any required filtering action after this initial external filter should be dimensioned such that they can be implemented on-chip whenever possible.

The external filters need to be connected to a matched load of 50  $\Omega$ . An attenuator at some point is desirable, as it can optimize SFDR. The maximum attainable SFDR is a function of several variables, including attenuation of the input signal, as will be discussed later in this section. Moreover, if the attenuation is put before any components that have a limited input-swing, it can increase the measurement range. Thus, attenuation very close to the input is preferrede.

Because the bandwidth of the system ranges from 0 GHz to 6 GHz, direct AD-conversion is not possible, which means that a frequency converter is required. Because the system is designed for high linearity, a switching mixer looks like a very good solution [60]. Switching mixers are operated using a square wave, so harmonics of the oscillator frequency  $f_{\rm LO}$  are created. These harmonics of  $f_{\rm LO}$  downconvert harmonic images of the desired frequency to be downconverted to the same frequency. Moreover, they introduce harmonics of the desired



Figure 4.1: An HR-mixer approximates an ideal sinewave by superposition of square waves (reproduced from [61]).

signal itself.

Several techniques exist to mitigate these problems. One concept is the use of Harmonic Rejection (HR)-mixers [61]. These mixers try to approximate the ideal situation, a pure sine wave, by a superposition of square waves. Superposition is achieved by putting several mixers in parallel and summing the results (see fig. 4.1). Which images and harmonics are removed depends on the combination used to approximate a pure sine wave.

A second concept is Polyphase Multipath (PM) [62, 63], which will be explained later in more detail. Unfortunately this technique does not remove all harmonics and images. The first image that is not canceled is situated at  $(W + 1)f_{\rm LO}$  before mixing, where W is the number of PM-paths used. These remaining images should be removed beforehand by the external filters at the input. It is thus possible to use a bank of external low-pass (or bandpass) filters, each with a cut-off frequency that is W times as high as the cut-off frequency of the previous filter. The canceling of harmonics and images is schematically depicted in fig. 4.2. Filtering and the cancellation of harmonics and images is a very common problem and not the focus of this thesis, and is therefore not fully worked out.

A block diagram of the resulting system design is shown in fig. 4.3. Each block will be discussed below, while some parts of the system are worked out in more detail in chapters 5 and 6. The difference between the two designs concerns the oscillator, which will be discussed later on.

#### 4.1.1 Antenna

In many RF-systems, an antenna is used to receive signals. For the SA it serves as a model for the signal source, because SAs are often directly connected to the Device Under Test (DUT). In RF-systems the wave-like nature of the signals cannot be neglected, and impedance matching is required to prevent reflection of the waves. It is customary practice to use an impedance of 50  $\Omega$ . The antenna or signal source is not part of the system, and is not further discussed.

#### 4.1.2 External filter bank

In the envisioned system, the filter bank consists of R low-pass filters, each having a cut-off frequency W times larger than the previous one. The first filter has a cut-off frequency at half the sampling frequency of the ADC, because then it can be immediately sampled without aliasing.<sup>1</sup>

Before a measurement, a specific filter is connected to the antenna and to the CMOSchip using some kind of switch. If the first filter is connected, no frequency conversion

 $<sup>^{1}</sup>$ The current design involves a low-IF receiver. In a zero-IF receiver with I/Q-mixers the bandwidth is equal to the sampling frequency.



(e) PM cancels many of the harmonics, but not all; they should be removed by an additional filter after conversion.

Figure 4.2: Schematic depiction of the cancellation of unwanted images and harmonics using filters and PM (for HR-mixers the idea is similar). The schematic picture is oversimplified as in the real world images and harmonics are not completely removed but only attenuated.



Figure 4.3: Two main designs of the correlation spectrum analyzer.

#### 4.1. SYSTEM DESIGN

is necessary and the input signal can be directly sampled. If any of the other filters is connected, frequency conversion is necessary, and the problem with images and harmonics comes into play. The filter removes all images > W before downconversion, while PM removes all images and harmonics  $\leq W$  'during' downconversion.

With a total bandwidth B of the SA, a sampling frequency  $f_s$  of the ADC, and a first harmonic W not removed by PM, the number of filters R required is

$$R = 1 + \left\lceil \log_W \frac{2B}{f_s} \right\rceil \tag{4.1}$$

which clearly illustrates the trade-off between the number of filters and the number of branches in PM.

Several types of external filters exist, such as Surface Acoustic Wave (SAW)-filters, Bulk Acoustic Wave (BAW)-filters, ceramic and crystal resonators, microstrip lines and filters consisting of discrete lumped elements (which do have a high Q). The exact operation of these filters is not important in this context, so it will not be discussed. The main reason to use the external filters is their higher frequency-selectivity, their much steeper roll-off, and their higher suppression in the stop band.

External filters typically have an Insertion Loss (IL) of several dB (ceramic 1 dB to 2 dB<sup>2</sup>, crystal 2 dB to 3 dB and SAW 3 dB<sup>3</sup>), which directly translates to NF. Because the external filters are expensive, they are not duplicated, and all noise added will be fully correlated. Hence, provided all other noise will be uncorrelated, this noise will ultimately limit the noise floor.

#### 4.1.3 Matching & Attenuation

With 65 nm technology, the maximum gate-oxide voltage is 1.2 V. With a threshold voltage of roughly 0.2 V, this leaves only about 1 V as input range. The maximum power then is equal to  $A^2/2R$ , where A is the maximum amplitude of a sine (0.5 V in this case). At  $R = 50 \Omega$  this is equal to 4 dBm. The attenuator needs to be designed such that higher input powers can also be handled with standard technology.

Moreover, the maximum attainable SFDR is a function of several variables, such as Second Order Input-referred Intermodulation Intercept Point (IP2), IP3, NF and oscillator phase noise [19]. Attenuation of the input signal increases IP2, IP3 and NF, but each with a different factor, see fig. 4.4. The option to choose between different attenuation levels allows optimization of the SFDR.

Using crosscorrelation, the NF can be lowered, and therefore the optimum attenuation changes with measurement time. Nevertheless, the need for variable attenuation remains.

To maximize linearity the use of only resistors and switches in the matching and attenuation network is attempted, consuming (virtually) no power. Chip area is very hard to predict, but is expected to be negligible compared to components such as the Voltage-Controlled Oscillator (VCO), the ADCs and the digital hardware.

#### 4.1.4 Mixer & PM

In a recent Master's thesis from Soer [60], a very linear mixer referred to as a Tayloe mixer was investigated and shown to be very linear. An implementation was made on chip in 65 nm technology with a supply of 1.2 V. The most recent worst-case figures are an IP3 of +11 dBm, an NF of 6.5 dB and a gain of 19 dB. The power consumption is 67 mW, of which around 15 mW is consumed by the buffers driving the switches. The other power consumer is the LNA at IF. The active chip area is less than 0.13 mm<sup>2</sup>: 0.04 mm<sup>2</sup> for the IQ-mixer, 0.08 mm<sup>2</sup> for the two IF-amplifiers and the remaining area for a clock-divider.

<sup>&</sup>lt;sup>2</sup>http://www.t-ceram.com/ceramic-filters-diplexers.htm

<sup>&</sup>lt;sup>3</sup>http://www.vanlong.com



Figure 4.4: The SFDR is a function of IP2, IP3, NF, oscillator phase noise and attenuation of the input signal (graph from the Excel-sheet referred to in [19]).

These figures will be published at the ISSCC of 2009. Linearity is severely limited by the LNA at IF. The Tayloe mixer itself in simulations showed an IP3 of +27 dBm. Because of the high linearity, this will be the frequency converter of choice.

The Tayloe mixer uses switches to perform the frequency conversion. Multiplication is therefore performed by a square wave with a certain duty cycle, resulting in many harmonics of the fundamental oscillator frequency. Only one of these harmonics yields the desired IF, while the others result in unwanted signal components. Traditionally these unwanted signal components are removed using filters, but for low-IF the closest unwanted signal component is so close to the desired signal component that on-chip filtering becomes impossible.

A recent development called PM [62, 63] (partly) solves this problem by canceling some of the harmonics. Canceling of harmonics is achieved by replicating the system to create several paths, applying a phase-shifted version of the input signal (or oscillator signal) to each of these paths, reverse the phase-shift at the output of each path and add all the signals. This process is schematically depicted in fig. 4.5. By making sure the first non-canceled harmonic is far enough from the desired IF, an on-chip filter can be used. Some additional techniques to suppress undesired images and harmonics are discussed in the recommendations.

#### 4.1.5 Oscillator

The oscillator generates a signal that can be used to operate the Tayloe mixer and the sampling moments of the ADC. In most systems, a quartz-based oscillator (which is an off-chip device) is used to generate a very stable reference frequency. It relies on piezo-electricity to transform mechanical resonance, determined by the shape of the crystal, into a voltage waveform. The oscillating frequency is temperature dependent. This temperature dependency is usually compensated, but some residual dependency always remains. With high absolute frequency requirements even this residual dependency results in too much inaccuracy, which is why many commercial SAs thermally stabilize the oscillators [64]. This in turn means that SAs require a warm-up time after power is switched on to meet the specifications listed in the datasheets.

Because many systems need a more tuneable frequency range than can be directly pro-



Figure 4.5: The principle of PM to cancel unwanted harmonics and images (reproduced from [63]).

vided by the crystal (which is almost always the case because their tuning range is just enough to compensate for loading effects), one usually employs a VCO in the form of an LCor ring-oscillator, connected with the crystal through a frequency divider in a Phase-Locked Loop (PLL) [65]. For the SA under design, the bandwidth ranges from 0 GHz to 6 GHz. It is thus necessary to be able to generate a frequency ranging from  $f_s$  to (6 GHz –  $f_s$ ).

Below 1 GHz, on-chip inductors are too large and suffer from too much internal resistance to be of practical interest. Above 1 GHz however, they provide a far more energy-efficient oscillator than can be provided by ring-oscillators, which have a  $Q \leq 1$  because there is no energy reuse. A better energy efficiency than that of ring-oscillators is necessary, because Rover's [4] phase noise requirement of -134 dBc/Hz at 1 MHz offset at an oscillator frequency of 1 GHz (which is also a typical specification of commercial SAs [19]) requires a power consumption of roughly 2.5 W for a ring-oscillator, while an LC-oscillator requires only 8 mW (see section B.7 for the derivation). These numbers only reflect the power required for the oscillator itself, and do not include power for the buffers. The power consumption of LC-oscillators increases if tunability is required [66].

The previous power estimates are based on the assumption that a 1 MHz offset is (just) outside of the PLL bandwidth. To a first-order approximation, the noise spectrum inside the PLL bandwidth is flat, while outside of this bandwidth it follows the plain phase noise of the VCO until it hits the white noise floor generated by components such as the buffer. This white noise floor can usually be neglected. Many cheap crystals run at a frequency of 10 MHz or 30 MHz, and the bandwidth of a PLL is usually set to roughly 10% of  $f_{\rm ref}$  for various reasons, which makes the assumption that the phase noise of the VCO itself should be -134 dBm/Hz at a 1 MHz offset justified.

For smaller RBW, the phase noise may be higher to achieve the same SFDR: an RBW



RBW / Offset Frequency [kHz]

Figure 4.6: The influence of phase noise on maximum attainable SFDR as a function of RBW or offset frequency.

of 10 kHz gives a phase noise requirement of -114 dBc/Hz at 10 kHz offset. In commercial SAs the SFDR usually increases for smaller RBW. Using a PLL this effect is achieved due to the flat noise spectrum within the PLL bandwidth. At some point the floor will start to rise due to flicker noise, and the SFDR will saturate.

#### Oscillator design proposal

An oscillator design is proposed and worked out to some extent to obtain a reasonable estimate of the required power consumption.<sup>4</sup> An LC-VCO tunable from 10 GHz to 12 GHz is designed and created in [67] and is shown in fig. 4.7. It uses 0.18  $\mu$ m CMOS-technology operating at a 2.2 V supply. The measured phase noise is -125.33 dBc/Hz at 1 MHz offset at a power consumption of 50 mW. The Figure-of-Merit (FoM) is -188 dBc/Hz, which is among the best available at these frequencies [66].

Before continuing, a few assumptions have to be made:

- The same specifications can be attained with a 65 nm process. This seems reasonable as [66] suggests performance improves with technology, but the system under design uses a lower supply voltage. A 1.2 V supply voltage can still easily handle the required overdrive voltages of the transistors used for providing negative resistance (see fig. 4.7).
- Phase noise can be scaled the same way as thermal noise: doubling the power consumption lowers the phase noise by 3 dB. This is more or less true for fixed oscillators, but at least questionable for tunable oscillators [66].
- Multiple oscillators can be safely put on a single chip without significant coupling. Because the entire SA will be integrated onto a single chip, the chip will be relatively large. This means the inductors can be placed far apart, which decreases mutual coupling significantly [68].

By making a second LC-VCO, tuneable from 8 GHz to 10 GHz, the total tuning range covers 8 GHz to 12 GHz. Both VCOs need to have -134 dBc/Hz phase noise at 1 MHz offset, which is 9 dB better than the implementation in [67]. The frequency will be divided by at least a factor two, which decreases the phase noise requirement by 6 dB. The net results

 $<sup>^{4}</sup>$ Another idea for an oscillator architecture, which is more power-efficient but may have some significant drawbacks, is presented in appendix C.



Figure 4.7: The tuneable LC-VCO (reproduced from [67]).

is that the power consumption needs to be two times higher, giving a total of  $P_{\text{VCO}} = 2 \cdot 2 \cdot 50 = 200 \text{ mW}$ . Note that these numbers include the buffer power.

In the case of crosscorrelation however, these specifications may be lowered. When a separate oscillator is used for both branches, any uncorrelated phase noise will be removed, again at the cost of longer measurement time. In that case it is extremely important that their frequencies are equal, because otherwise both the signal and the noise will be correlated away (unless a fixed frequency difference is corrected for in the digital domain, see section 4.2). This can be done e.g. by locking them to some external crystal through the use of separate PLLs. In this case, the only correlated noise comes directly from the crystal, which can be really low [64].

Using integer frequency dividers one can generate any frequency desired from the 8 GHz to 12 GHz combined tuning range of the VCOs. With some 'puzzle solving' a way was found to do this using only two divide-by-2 and four divide-by-3 integer dividers. This is important because higher prime numbers require more logic and are more difficult to implement operating at rates of 12 GHz. An overview of how to acquire the full tuning range is shown in table 4.1. It should not be a problem to let the control-unit set the right connections between the frequency dividers.

The architectures of divide-by-2 and divide-by-3 frequency dividers are shown in fig. 4.8. Another implementation is given in [69]. Note that all of these architectures produce an output signal with a 50% duty cycle. Using Current-Mode Logic (CML), these circuits can be made very fast, even up to 40 GHz using 80 nm CMOS [70], so it should not be a problem to operate them at 12 GHz using 65 nm CMOS.

An upper bound on the power consumption of the frequency dividers will be calculated to comply to the phase noise requirement. This upper bound gives a value much lower than the power required for the VCO, which means that a more detailed analysis is not required.

The total phase noise introduced by the cascade of frequency dividers should not exceed -134 dBc/Hz at an offset of 1 MHz. The phase noise of frequency dividers has a flat spectrum from 0 to  $f_{\text{out}}/2$  [71]. Since maximally four dividers are cascaded, the phase noise of each divider should not exceed -140 dBc/Hz. Based on table 4.1, one divide-by-2 and one divide-by-3 is required to operate at an input frequency of 12 GHz, one divide-by-2 to operate at 6 GHz and one divide-by-3 each to operate at 4 GHz,  $\frac{4}{3}$  GHz and  $\frac{4}{9}$  GHz.

Factor	Div-by-2	Div-by-3	$f_{\rm low}$ [GHz]	$f_{\rm high}  [{\rm GHz}]$
1	0	0	8.000	12.000
2	1	0	4.000	6.000
3	0	1	2.667	4.000
4	2	0	2.000	3.000
6	1	1	1.333	2.000
9	0	2	0.889	1.333
12	2	1	0.667	1.000
18	1	2	0.444	0.667
27	0	3	0.296	0.444
36	2	2	0.222	0.333
54	1	3	0.148	0.222
81	0	4	0.099	0.148

Table 4.1: Frequency generation using a VCO, tunable between 8 GHz and 12 GHz, and integer frequency dividers.

The current consumption per latch can be calculated using [71]

$$\mathcal{L}_W = 8\pi^2 \left( 1 + \frac{\gamma}{\alpha} + \frac{\gamma_T g_{m_T} R_L}{2\alpha_T} \right) \frac{kTC_L}{I^2} f_{\text{out}} \, \left[ \text{dBc/Hz} \right]$$
(4.2)

 $C_L$  is taken as 20 fF, just as in [69]. Based on a conservative value of 2 for the noise factor of a transistor, a voltage swing of 0.6 V and an overdrive voltage of 0.2 V [73], the factor between parentheses is roughly equal to (1 + 2 + 3) = 6. From this equation and a -140 dBc/Hz phase noise requirement follows

$$I_{\rm div-by-2} = 2.75 \times 10^{-9} \sqrt{f_{\rm out}} \,\,[{\rm A}]$$
(4.3)

This means that the latches in the divide-by-2 frequency divider operating at an input frequency of 12 GHz require a current of 213  $\mu$ A. Since a divide-by-2 has two latches in cascade, the total power consumption at a supply voltage of 1.2 V becomes 0.51 mW (note that this is a conservative value because the first latch does not contribute to phase noise if it obeys the setup- and hold-times of the second latch).

Divide-by-3 frequency dividers also have additional logic gates. To be on the safe side, a divide-by-3 is treated as consisting of six latches. The longest path from clock to output consists of three 'latches'. Because phase noise adds incoherently, the phase noise of each latch needs to be  $10 \log 3 \approx 5$  dB lower, or equivalently -145 dBc/Hz. Hence

$$I_{\rm div-by-3} = 3.46 \times 10^{-9} \sqrt{f_{\rm out}} \,\,[{\rm A}]$$

$$\tag{4.4}$$

The power consumption of the divide-by-3 running at an input frequency of 12 GHz has a power consumption 1.58 mW.

The total power consumption of the frequency divider then becomes (0.51+0.37)+(1.58+0.91+0.53+0.30) = 4.2 mW. All these numbers are upperbound numbers, especially for the divide-by-3 dividers, because not all components contribute in the same amount to the phase noise, and can therefore run at a lower current. Nevertheless, the total power consumption of the frequency dividers is insignificant compared to the VCO.<sup>5</sup>

The chip area is severely dominated by the inductors and does not scale significantly with technology. In [67] the VCO occupies an area of  $0.67 \text{ mm}^2$ . In the current design two

<sup>&</sup>lt;sup>5</sup>Note that it is assumed that flicker noise is not an issue, but this requires verification.



Figure 4.8: Architecture for integer frequency dividers.

VCOs are required to cover the whole tuning range, occupying a total area of 1.33 mm<sup>2</sup>. This might be reduced by placing parts of the VCO underneath the inductors [66]. The size of the frequency divider network is negligible.

#### 4.1.6 IF-circuitry

The IF-circuitry is the circuitry between the output of the mixer and the input of the ADC. It should be made very linear, and because it is assumed that the IF-frequencies only range up to 100 MHz, a switched-capacitor implementation seems most promising.

In this design the RF-frontend does not amplify the incoming signals, so amplification should be done at IF. Harmonics introduced by the switching mixer should be filtered out using a Low-Pass Filter (LPF). Finally, the interface to the ADC should contain an SH to provide a steady signal.

It is hard to estimate the power consumption, especially because the required gain (range) of the LNA and the implementation of the LNA are not known. Soer [60] found a power consumption of 50 mW for the LNA at IF. The filter might require some extra power, as well as the interface to the ADC. Therefore a total power consumption of 100 mW is assumed.

The area occupied by the two IF-amplifiers of the test chip designed by Soer is 0.08 mm<sup>2</sup>. A conservative estimate is a total area of twice this size: 0.16 mm<sup>2</sup>.

## 4.1.7 Analog-to-Digital Converter

The ADC will convert the analog input signal to a digital output. The effects of quantization have been elaborately discussed in chapter 3, leading to the conclusion that 9 bits is required for an SFDR of 70 dB. It is better to have some extra margin, especially because the ADC is not the only non-linear factor. Also, it may not be possible using automatic gain control to fully utilize the input range of the ADC. Therefore 10 bits is the resolution opted for.

A pipelined ADC-architecture gives the required resolution at a sample rate of 200 MS/s, see [48]. Based on [74] the FoM at 10 Effective Number of Bits (ENOB) is about 0.2 pJ per conversion. At a sample rate of 200 MS/s a power consumption of 40 mW is expected.

An ADC with specifications very close to what is desired (10 bits, 200 MS/s) is discussed in [75], with a power consumption of 61 mW, so it seems like a reasonable assumption.

Although not completely fair because it is a different architecture, an on the chip area can be obtained using another recently designed ADC [76]. The ENOB is around 10 bits for sample rates up to 200 MS/s, and the FoM is 0.6 pJ per conversion. The chip area is 1.6 mm<sup>2</sup> in a 0.13  $\mu$ m CMOS-process, but will probably not scale much in a 65 nm CMOS-process due to matching and noise requirements. It uses interleaving with 16 identical time-multiplexed ADCs, all using the same time-interleaved track & hold. For a sample rate of 200 MS/s, the 16 channels may be reduced to 1, 2 or 3 channels. Based on the chip photograph, this would bring the total area down by about 50%, resulting in approximately 0.8 mm<sup>2</sup>. The chip area of the ADC in [75] is 1.0 mm<sup>2</sup>, so again the value thus obtained seems like a reasonable assumption.

## 4.1.8 Correlator

After AD-conversion, digital signal processing is required to obtain the spectrum. The spectrum is to be obtained using crosscorrelation. In order to maintain a logical flow of information in this thesis, the different digital correlators, and the estimation on power consumption and chip area, will be discussed in chapter 6.

## 4.1.9 Control

The control unit will control all the analog and digital components to comply to demands from a user or running application. It is most likely a fully digital piece of hardware that outputs the correct signals to select the external filter, the attenuator-setting, the oscillator frequency for downconversion, the sampling rate of the ADC, the number of correlation samples to calculate and the moments to start and stop measuring or calculating. It may also include methods for calibrating the different parts. It will receive the desired functional behavior of the block from an external source.

During a measurement the control unit does not have to do much, so its power consumption will form a negligible contribution to the total power consumption. It is assumed that most of the control operations can be performed by the same core that handles the digital correlation, so (virtually) no additional chip area will be needed.

## 4.2 Cascading suppression mechanisms

Based on Rover's research [4], a low-IF topology is chosen. The high requirements on SFDR however could become a problem. Based on all the non-idealities like feedthrough, mismatch and non-ideal filters (for simplicity it is assumed that each mechanism gives 40 dB suppression, which is achievable [20, 23, 77]), the desired 70 dB suppression of all undesired signals and spurs can never be met in one step, but a cascade of suppression mechanisms is needed to achieve the desired SFDR.

For a 40 dB suppression per mechanism, two suppression mechanisms in cascade are enough to allow the SFDR to reach 70 dB. We will take a look at fig. 4.3 to determine which parts may become a problem.

The external filters suppress many images by 40 dB, because of feedthrough and nonideal frequency characteristics. Because they are low-pass filters, they will not suppress the images close to the desired frequencies and the signals below the desired frequencies. Harmonic distortion in external filters is negligible.

The mixer suppresses many harmonics and images by only a few dB, as the harmonics of the square wave function only slowly decay in amplitude. For simplicity this small suppression is ignored as it is insignificant for the problem at hand. The mixer converts the images to baseband. Because of feedthrough, signals present at IF before frequency conversion will still be present at IF after frequency conversion, although suppressed by 40 dB.

#### 4.3. POWER CONSUMPTION AND CHIP AREA

The PM-part suppresses many harmonics and images by 40 dB, but not all. The phaseshifters will have feedthrough, but this feedthrough is not phase-shifted. The second set of phase-shifters will suppress them by another 40 dB, either by feedthrough or PM (because in that case they are phase-shifted only once and thus canceled by the summation).

This means that at IF the following signals are present:

- 1. The desired signal (unsuppressed)
- 2. Images suppressed by external filters (40 dB suppressed)
- 3. Images suppressed by PM (40 dB suppressed)
- 4. Harmonics suppressed by PM (40 dB suppressed)
- 5. Images suppressed by both external filters and PM (80 dB suppressed)
- 6. Signals originally present at IF (80 dB suppressed)

Of all these signals the ones that are only 40 dB suppressed form a problem.

An on-chip filter can suppress the harmonics another 40 dB if they are located far enough from IF, which need not be the case if the oscillator frequency is just above the cut-off frequency of the first external filter. This would be enough to tackle item 4, but would leave items 2 and 3 at 40 dB suppression, as the images are already at baseband and cannot be distinguished from the original signal.

A very interesting and very recent development by Moseley [78] is to use crosscorrelation to remove the images that are converted to baseband by harmonic downmixing. This will suppress *all* harmonics and images. It uses virtually the same structure and same ideas as in this design, and so it may be possible to integrate this solution at virtually no extra cost. The idea is to have the second VCO operate at a slightly different frequency with offset  $\Delta f$ , such that the harmonic images in one path are not equal to the images in the other path. By digitally correcting the offset frequency through simple multiplication with a sine wave, the images in both paths will be uncorrelated and only show up as noise, which means that with long enough measurement time they can be completely removed.

## 4.3 Power consumption and chip area

In the analog world there is a tradeoff between power consumption, noise, linearity, bandwidth, speed, gain and voltage headroom. Virtually all of the power consumption goes into biasing of the transistors. For a desired SNR the power consumption is more or less technology independent [9]. Taking into account only thermal noise, the SNR can be increased by 3 dB while keeping the same bandwidth if all capacitances are doubled and all resistances are halved.<sup>6</sup> This leads to a doubling in analog power consumption and an increase of chip area. This scaling cannot be applied to the ADCs as they still have to deliver the same resolution.

Since crosscorrelation has two paths, whereas the standard SA needs only one path, this method requires twice as much power and roughly twice as much area under the assumption that the amount of noise in each branch needs to be kept equal to the standard case. Under the assumption that for equal power consumption the noise in each branch is increased by 3 dB, the measurement time needs to be increased by a factor 4 to get rid of this extra 3 dB of noise.

However, power consumption alone does not provide the whole picture if the analog circuit can be switched off after a measurement is complete. Note that when the noise level goes up by 3 dB and in the end the same requirement holds for the measurement result, the

<sup>&</sup>lt;sup>6</sup>Several factors limit this approach, such as the maximum currents transistors can handle before they break down, and the maximum current certain topologies can handle before they run out of voltage head-room.

total energy consumption of the analog part increases by a factor of 2 and of the digital part by a factor of 4. The battery in a battery-powered device will be drained faster in terms of number of measurements. So if energy consumption is the main limitation, it would be best to lower the noise as much as possible. This is an important consideration in a more detailed design.

## 4.4 Measurement time

A measurement time that is 100 times longer than without correlation does not say very much as often only the absolute measurement time is relevant. An increase from 1  $\mu$ s to 100  $\mu$ s will in many applications be tolerable, while an increase from 1 s to 100 s may not be tolerable. Some applications mentioned in chapter 1 will be used to illustrate this, followed by some other applications.

#### 4.4.1 Cognitive Radio

For CR, a draft exists to define the standard [79], which states that signals as low as -116 dBm need to be detected in a 6 MHz bandwidth during a fine scan, which is allowed to take roughly 25 ms. With a NF of 0 dB, the noise level in a 6 MHz bandwidth will be -114 dB, so it will be very difficult to detect such signals and in this respect the proposed standard seems somewhat odd. The paper focuses mainly on the TV-bands (54 MHz to 800 MHz) in combination with base stations, while other CR-designs consider ad-hoc connections.

In the case of ad-hoc connections, the frequency band to scan goes up to 6 GHz [16], which is rather wide, such that the scanning-FFT based spectrum analysis as suggested in [15] may be a good choice. Assuming the ADCs run at 20 MHz for this application, roughly 500 analyses need to be made, such that the total time per analysis is in the order of 2 ms, which allows 40.000 samples to be taken.<sup>7</sup> If a frequency resolution of 10 kHz is required (which is about the minimum channel width used in any modulation scheme), a 1000-points DFT needs to be calculated. Using all of the 40.000 samples, instead of the minimum of 1000 samples, increases the measurement time by a factor 40, but also decreases the noise floor by 8 dB.

While transmitting, CR needs to respond relatively fast to changes in the spectrum, because interference with licensed users is not allowed. The large distances (> 100 km) that can be covered [79] also makes collisions between CR-users possible. The maximum interference time allowed, as suggested in [79], is 2 s. Transmission needs to be interrupted in order to be able to scan, so to achieve an efficient spectral usage, the scanning time should be much less than transmission time. Because now only the range in which the radio is transmitting needs to be scanned, this should not pose any problems.

In any case, the only 'danger' is a false positive on a frequency band being used, in which case no real harm is being done. When the SA gives a false positive (i.e. there is apparently a lot of noise in this band), it is very likely that the radio itself will also have difficulty using this frequency range, because parts of the front-end will most likely be shared by the radio and the SA. A false positive in this context may not be such a bad thing at all.

## 4.4.2 Built-in Self-Testing

Although on-chip frequencies already range up to 60 GHz [80], most signals still are below 6 GHz, which allows the envisioned SA to measure them. In case BIST uses a built-in signal generator, measurement time is allowed to be rather long, as long as that part of the chip is not needed for the application it is built for. If it only uses true application signals, it

 $<sup>^7\</sup>mathrm{This}$  is not entirely true. Filters etc. require some time to settle, so the amount of useful samples will be lower.

#### 4.5. CONCLUSIONS

completely depends on the application and the type of signal. Taking CR as an example, the signals it sends will probably use Orthogonal Frequency Division Multiplexing (OFDM) [15], which may be sent in bursts. With time gating, measurement time may also be rather long, as samples are taken only during transmission.

#### 4.4.3 Phase noise measurements

When measuring phase noise of an oscillator using an SA, the phase noise of the VCO in the SA should be significantly lower than the phase noise of the oscillator to be measured. If two frequency-locked VCOs are used in the correlation SA, correlation will lower the effective phase noise of the internal VCOs, allowing detection of virtually arbitrarily low phase noise of the DUT. Measurement time in this case is only limited to days or maybe even weeks, so very sensitive phase noise measurements may be obtained. Note that phase-locking to the DUT, such as done in [40], allows faster reduction of the noise.

#### 4.4.4 Linearity measurements

Just like measuring phase noise requires an SA with extremely low noise phase noise, measuring the linearity of a circuit requires an SA that is significantly more linear than the DUT. The linearity of the SA can be artifically improved by using an external attenuator, but this of course means that the noise level goes up. Using crosscorrelation, this noise can then be lowered. A requirement then becomes that the attenuation is performed separately in each path, because otherwise it would introduce noise correlated in both paths. Again measurement time is only limited by practical constraints.

## 4.5 Conclusions

Signals significantly below the noise floor will not be detected by a regular SA because of measurement errors and a not-completely-white noise floor. Although these effects will also be present in crosscorrelation, crosscorrelation lowers the noise floor itself, allowing detection of smaller signals. Therefore, a high-level system design of an SA is proposed that uses crosscorrelation.

The design uses low-IF as suggested by Rovers [4]. An external filterbank with logarithmically distributed cut-off frequencies is placed directly after the antenna to prevent crossing the chip-boundary more than once. Impedance matching is required to let the filters operate properly. Variable attenuation is put close to the input to lower linearity requirements of active components. Mixing the RF-signal to IF is done before any amplification, again to improve linearity.

Since highly linear mixers are switching, mixing is also performed with many harmonics of the oscillator signal. Image rejection is required to suppress this undesired effect. Some images are rejected by the external filters at the input; the other images are removed by techniques such as HR-mixing and/or PM. For both methods there is an important trade-off between the number of external filters on one hand and the number of paths in PM or the approximation of the sinusoid in HR-mixing on the other hand. Harmonics of the desired signal are removed by PM or HR-mixing in combination with on-chip or off-chip filtering. Non-idealities in all mechanisms makes a cascade of two techniques necessary to achieve an SFDR of 70 dB.

The main power consumer of the analog front-end is the VCO. The suggested implementation uses two 20%-tuneable LC-oscillators with integer frequency dividers to cover the whole frequency range from 100 MHz to 6 GHz, requiring an estimated 0.2 W.

Decreasing the noise in the analog frontend by 3 dB increases the analog power consumption by a factor of 2 (except for the ADCs), while the digital power consumption remains the same. Decreasing the noise by 3 dB using correlation does not change the analog and digital

power consumption, but increases measurement time by a factor of 4. It is thus advisable to decrease the noise in the analog frontend by power scaling whenever possible.

To allow correlation to reduce the noise, as much noise as possible should be uncorrelated. For component noise this should not be a problem. Correlated noise can originate from ground bounce and power supply, so the analog parts should be designed with a high rejection ratio against these variations.

A solution to lower power consumption is to use two identical VCOs locked in frequency (using for example another PLL). Their phase noise would then be uncorrelated and can be reduced through correlation. This could also lower the phase noise requirements and hence the power consumption of the oscillators, but of course at the cost of increase in measurement time.

Using correlation as a means to lower the noise floor allows the SA to be used as a very sensitive phase noise and linearity measurement instrument, because in these situations there is no strict limit on measurement time.

## 4.6 Recommendations

There are several possible improvements to the current design, which are discussed here.

## 4.6.1 ADC resolution

The required ADC-resolution for a general-purpose SA to obtain an SFDR of 70 dB is 10 bits. For certain applications one may need less, and not just for reasons of SFDR. Kokkeler [43] points out that coarse quantization does not give an accurate representation of a signal, but does show its characteristics. This may be useful in CR. One could include additional ADCs or reuse the existing ones with a much lower power consumption. Similarly the digital hardware can become much less power-consuming. A similar approach is discussed in [81].

Bitstream processing is another idea to lower the requirements on digital hardware [82]. The operations are performed on the low-resolution output of e.g. a  $\Sigma\Delta$ -modulator instead of higher resolution words as in regular ADCs. More research is required to determine the effects of bitstream processing on properties such as the SFDR and the complexity and power consumption of the digital hardware.

#### 4.6.2 Lowering VCO power consumption

The phase noise requirement of -134 dBm/Hz at a 1 MHz offset results in a relatively high power consumption of the VCO because the frequency offset is approximately at the outer edge of the PLL bandwidth. This bandwidth is roughly 10% of the reference frequency provided by an external crystal. A crystal with a higher reference frequency can therefore lower the phase noise requirements of the oscillator: phase noise has a  $1/f^2$  PSD-dependency, so (to a first-order approximation) if the PLL bandwidth can be increased by a factor of 10, the power consumption of the VCO can go down by a factor of 100.

### 4.6.3 Regular spectrum analysis

Being able to lower the noise floor is a good thing, but it is not always necessary. Sometimes it is more important to quickly measure a quantity, or it is desirable to be able to see the respective phases of components. With correlation, phase information is lost because the input signal is directly transformed from the time domain to the power domain.

It would be desirable to have the option of performing regular spectrum analysis using a single stream of data, where the FFT of the data gives amplitude and phase information. The expected costs are low, because all of the required components are already needed in crosscorrelation measurements. It may be possible to perform two measurements at once at

#### 4.6. RECOMMENDATIONS

two different VCO-frequencies (when two separate VCOs are used), or it may be possible to shut down one branch to save power.

When two VCOs are present for correlation, it might be possible to use them as quadrature oscillators for regular spectrum analysis or for any other application that requires it.

#### 4.6.4 Correlation with one measurement path

Instead of using two measurement paths it may also be possible to use one measurement path. The idea is to interleave the single measurement path in time to mimic two paths.

One issue is that the signal to be measured needs to be highly correlated in time or the correlation will yield meaningless results. A signal that obeys this correlation in time is a sinusoid, while a signal that does not obey this correlation in time is white noise. This idea may not work for general-purpose usage, but it could come in handy for certain applications where there simply is not enough chip area or power available to allow two measurement paths.

### 4.6.5 Suppressing harmonics and harmonic images

As was discussed in section 4.2, direct downconversion cannot guarantee an SFDR of 70 dB, simply because some undesired signals cannot be attenuated enough. One possible solution mentioned was the frequency-offset method from Moseley [78]. As commercial SAs generally implement a superheterodyne architecture with high-IF, it might also be interesting to see if that could work, even though it may require additional off-chip filters [23].

#### 4.6.6 Combining correlation and PM

It is possible to combine correlation and PM by giving one branch one extra path for PM as shown in fig. 4.9. This means different harmonics and images are canceled by the paths; some are canceled by both, others are canceled by only one branch, while the first image or harmonic that is not canceled by either is much further away in frequency than in the case where both branches have an equal number of paths. In the case where the number of paths  $p_1$  and  $p_2$  in both brances are coprime, i.e. they share no common factor other than 1, the first non-canceled harmonic or image is  $1 + p_1 p_2$ .

Because of mismatch, PM only suppresses images and harmonics by roughly 40 dB, or equivalently, a factor 100 in voltage. When the image or harmonic is suppressed by only one branch, it will still be present at the output with a 20 dB suppression (in one correlation branch it is suppressed by a factor 100, while in the other branch it 'suppressed' by a factor 1, which after multiplication gives a suppression by a factor 100). In combination with other suppression mechanisms, it may be possible to suppress *all* images and harmonics by the required 70 dB. The remaining images and harmonics are so far away that filtering should not be a problem anymore.

In either case, harmonics and images suppressed by only one branch will act as noise in the other branch. They will appear as noise at their specific frequencies after frequency conversion. A longer measurement time will be required to remove the effect altogether.

#### 4.6.7 Combining PM and frequency conversion

Mixers are almost ideal phase-shifters [62]. If the Tayloe mixer is used as one phase-shifter, then another mixer is required for reverse phase-shifting because the result is again relatively wideband. The problem is that this will give another frequency conversion. Therefore it may be better to have the Tayloe mixer convert to an IF that is not at baseband. The second mixer then converts the signal to baseband. This solution might also work for (additional) suppression of images and harmonics.



Figure 4.9: Correlation and the PM-technique form a unique opportunity to suppress a lot of images and harmonics at very little cost.

## Chapter 5

## Analog frontend

In order to obtain a good estimate on the possible linearity (IP3) in combination with the NF and power consumption of the system, an initial implementation of the analog frontend was made. Unfortunately, due to lack of time, not all functions could be implemented. It was conjectured that the High Frequency (HF)-part up to and including the frequency conversion will dominate linearity restrictions and NF, because it is more wideband and operating at higher frequencies than the IF-part. Therefore only the HF-part is discussed in this chapter.

The signal to be measured first goes through an external filter before it enters the chip. Since external filters generally require a 50  $\Omega$  matched load, the first stage on the chip should be an impedance match to the filter. A tunable attenuation is then required to optimize the SFDR (see fig. 4.4). At the same time, it is desirable to increase the measurement range of the SA to allow a larger range of signals to be measured. After impedance matching and attenuation, frequency conversion is performed. Because the proposed design combines these functions, they will be discussed as being one. For simplicity, this circuit will be referred to as the attenuator.

At IF, (variable) amplification is required, as well as filtering and an SH circuit to be able to enter the ADC. This is not discussed in this thesis.

## 5.1 Attenuator

Using crosscorrelation, the NF can be lowered, and hence the optimum attenuation in terms of SFDR changes with measurement time. Nevertheless, variable attenuation is still needed. If the circuit is designed properly, it may also improve the measurement range of the SA.

In ADCs and DACs a well-known design principle is the R–2R-ladder, which is a regular structure consisting of resistors of  $R \Omega$  and resistors of  $2R \Omega$ . This structure is shown in fig. 5.1. It has the nice property that at every node the impedance seen looking away from the input is  $R \Omega$ . The effect is that the voltage at each node is half the voltage of the node one step higher. In power this means each step has an attenuation of 6 dB.

In theory, this structure can be extended to arbitrary length, but in practice mismatch between the resistor values limit the applicability for DACs and ADCs. Moreover, the standard R–2R-ladder assumes each branch to be connected to (virtual) ground. For an SA, the resistor mismatches and a load unequal to 0 are not a problem. For absolute accuracy, the attenuation at each node needs to be calibrated anyway (e.g. by applying a reference voltage, measuring the attenuation and storing the value in a register) to correct for tolerances in the production process. Nevertheless, this structure can be used to provide a stepwise attenuation with steps of approximately 6 dB.

A schematic depiction of the attenuator is shown in fig. 5.2. Since it uses only resistors and switches, it is expected to be extremely linear. The attenuator will be connected to



Figure 5.1: The R–2R-ladder has a very regular structure and can be used for ADCs as well as DACs.



Figure 5.2: The attenuator works by connecting one of the branches to the load, and the other branches to ground.

the load, which in this case is a Tayloe mixer (the Tayloe mixer will be discussed on page 63). This means that one of the branches will be connected to the load, while the others will be connected to ground. This is symbolized by the switches. During a measurement, this assignment is static, i.e. it does not change. Since a Tayloe mixer has an almost infinite input impedance [60], the input impedance of the R–2R-ladder is not  $R \Omega$ , and depends on the branch connected to the load.

To simplify analysis it is assumed that the Tayloe mixer has an infinite impedance. Numbering branches from top to bottom, starting at 1, the branch connected to the load is denoted by s. The impedance looking into the attenuator is denoted with  $Z_{in}[s]$ , see fig. 5.3.

It can then be observed that  $Z_{in}[1]$  of the attenuator is  $2R \Omega$ . When the infinite load is connected to the lowest branch, the input impedance is only slightly larger than  $R \Omega$ . In the general case, the input impedance varies between  $R \Omega$  and  $2R \Omega$ . The input impedance can be calculated using the recursion relation that is obtained directly from figs. 5.2 and 5.3



Figure 5.3: Definition of input impedance  $Z_{in}[n]$  and node voltage  $V_{k,n}$ .

with  $Z_L \to \infty$ :

$$Z_{\rm in}[1] = 2R \ [\Omega]$$
  
$$Z_{\rm in}[n] = 2R \frac{R + Z_{\rm in}[n-1]}{3R + Z_{\rm in}[n-1]} \ [\Omega]$$

yielding as general solution (see section B.8)

$$Z_{\rm in}[n] = \frac{4^n + 2}{4^n - 1} R \ [\Omega] \tag{5.1}$$

which quickly converges to  $R \Omega$ .

The attenuation depends on the branch the load is connected to. The voltage  $V_{k,n}$  is defined as the voltage on node k when the load is connected to branch n, see fig. 5.3. From fig. 5.3 follows

$$V_{1,n} = \frac{Z_{in}[n]}{Z_{in}[n] + Z_A} V_{in} \ [V]$$
$$V_{k,n} = \frac{Z_{in}[n-k+1]}{Z_{in}[n-k+1] + R} V_{k-1,n} \ [V]$$

where  $Z_A$  is the antenna impedance and  $V_{in}$  the voltage received by the antenna. The general solution then is

$$V_{k,n} = \frac{\prod_{i=0}^{k-1} Z_{in}[n-i]}{(Z_{in}[n] + Z_A) \prod_{i=1}^{k-1} (Z_{in}[n-i] + R)} V_{in} \quad [V]$$
(5.2)

and the attenuation for branch n can then be found by calculating  $V_{n,n}$ .

For impedance matching, a general design rule is that the reflection coefficient  $S_{11}$  should be less than -10 dB [58, 83, 84]. The scattering parameter can be calculated using

$$S_{11} = \left| \frac{Z_{\rm in} - Z_{\rm out}}{Z_{\rm in} + Z_{\rm out}} \right| \tag{5.3}$$

where  $Z_{\rm out}$  is the output impedance of the antenna/filter and  $Z_{\rm in}$  is the input impedance of the attenuator. With  $Z_{\rm out} = 50 \ \Omega$  and  $S_{11} < -10 \ \text{dB}$ , it follows  $26 \ \Omega \le Z_{\rm in} \le 96 \ \Omega$ .



Figure 5.4: Nominal  $S_{11}$  for  $R = 50 \ \Omega$  and  $R = 35 \ \Omega$  and whole range of possible  $S_{11}$ -values if process variations can give a difference of 25%.  $S_{11}$  should be < -10 dB which makes  $R = 50 \ \Omega$  a bad choice.



Figure 5.5: Input impedance and gain of the attenuator with  $R = 35 \Omega$  and  $R = 50 \Omega$  as a function of the selected branch (without mixer but with infinite load).

To allow for mismatch and the effect of absolute offsets introduced by the production process, it is desirable to design  $Z_{\rm in}$  to be as close to 50  $\Omega$  as possible, where  $R \ \Omega \leq Z_{\rm in} \leq 2R \ \Omega$ . Hence, ideally  $R = 50/\sqrt{2} \approx 35 \ \Omega$ . With this value, for any load impedance connected to any branch (of course, all the other branches need to be connected to ground),  $S_{11} < -10$  dB. Without mismatch and an infinite load,  $S_{11} < -15$  dB, while even a 25% change in the resistor-values still keeps  $S_{11}$  below -10 dB, see fig. 5.4.

Figure 5.5 shows the input impedance and attenuation of the attenuator for the first six branches using eqs. (5.1) and (5.2) for  $R = 35 \Omega$  and  $R = 50 \Omega$ . The attenuation is calculated as  $-20 \log(V_{n,n}/V_{in}) + 6$ , where the term (+6) comes from the definition of Conversion Gain (CG) which is defined in terms of power and not voltage. Perfect impedance matching transfers all power, and gives a CG of 0 dB. At the same time the voltage is halved, so in terms of voltage one loses 6 dB. From fig. 5.5 it can be observed that the price paid for impedance matching is a slightly lower gain.



Figure 5.6: Principle of operation of a Tayloe mixer. The switch is operated with a 25% duty cycle.

Table 5.1: Calculated input impedance  $Z_{in}$ , resistor network gain (RNG) and total conversion gain (CG) of the designed RF-frontend.

Branch	$Z_{\rm in} \left[ \Omega \right]$	RNG $[dB]$	CG [dB]
1	70.0	1.32	0.41
2	42.0	-4.33	-5.24
3	36.7	-10.26	-11.17
4	35.4	-16.25	-17.16
5	35.1	-22.27	-23.18
6	35.0	-28.29	-29.20

#### 5.1.1 Tayloe mixer

A mixer is a device that performs frequency translation. Many implementations achieve this function by either passing or blocking the input. The fraction of time per period the signal is passed is called the duty cycle d.

The Tayloe mixer is extensively discussed in [60], and is a mixer that operates with a duty cycle of 25% ( $d = \frac{1}{4}$ ) in combination with an RC-time, which makes it a specific case of a switching mixer. The resistance R is the resistance seen from the capacitor, and C is the capacitor value itself. To differentiate between this R and the R used for the resistor network, the resistance seen from the capacitor is referred to as  $R_C$ , see fig. 5.6. Because the voltage difference depends on the entire history of the input signal (the Tayloe mixer is not stateless), a closed form expression of the transfer is very difficult to obtain. Soer found an expression in the frequency domain using Ström & Signell theory for Linear Periodically Time-Variant (LPTV) systems [85], from which some of the results are used here.

The RC-time of the Tayloe mixer acts as an LPF, with a cut-off frequency of [60]

$$f_{-3\mathrm{dB}} = \frac{d}{2\pi RC} \ [\mathrm{Hz}] \tag{5.4}$$

Note that this differs by a factor of two compared to [60] because of different definitions of R and C. The expression is only valid in case  $f_{-3dB} \ll f_{LO}$ .

Soer plotted the NF and conversion gain as a function of d for this type of mixer, which are reproduced in fig. 5.7. The specific case of  $d = \frac{1}{4}$  which defines the Tayloe mixer yields a good balance between Conversion Loss (CL) (0.9 dB) and NF (3.9 dB). CL is important for the total NF of the SA (because of the noise added by the IF-part). Together with the attenuation, the total calculated CL of this RF-frontend is shown in table 5.1.

Note that the NF shown in fig. 5.7 is for a balanced switching mixer. This balancing removes all even harmonics, so without balancing the NF will be 3 dB higher (see section B.9). Similarly, with I/Q-mixing, the undesired sidebands will be removed, lowering the NF by 3 dB.



Figure 5.7: Conversion loss and NF of a balanced switching mixer as a function of the duty cycle (reproduced from [60]).



Figure 5.8: The implemented RF-frontend combining the impedance match, attenuator and Tayloe mixer.

## 5.2 Circuit implementation

Requirements the circuit has to meet for any attenuation level are a frequency range of 0 GHz – 6 GHz and  $S_{11} < -10$  dB. An *RC*-bandwidth of 100 MHz is chosen because a sampling rate of 200 MS/s for the ADC seems like a reasonable speed (see page 51). For simplicity the circuit designed is unbalanced and does not use I/Q. Because the attenuator uses switches to connect either to the load or to ground, and the Tayloe mixer uses a switch for blocking or passing the signal, they can be conveniently combined in one device. The schematic is shown in fig. 5.8 and in this specific implementation has five branches, where the horizontal transistors  $A_1$  to  $A_5$  are the combined devices. The vertical transistors  $B_1$  to  $B_5$  connect their branch to ground if the capacitor C is not connected to that branch.

The oscillator-signal is applied to switch  $A_n$ , where  $n \in \{1, 2, 3, 4, 5\}$ , and the gate of  $B_n$
is connected to ground. The gates of switches  $A_j$ ,  $j \neq n$ , are connected to ground, and the gates of  $B_j$  are connected to  $V_{\text{DD}}$  (in the simulation this is implemented using some simple Verilog-A code).

The resistors with a value of  $2R \ \Omega$  shown in fig. 5.2 have been split up into three contributors. One is the switch to ground, one is a resistor (denoted with subscript *a*) in series with the mixing switch, and one is a resistor (denoted with subscript *b*) in series with the switch to ground. This allows correction for the different impedances seen from the load capacitor looking into any of the branches. Keeping the impedances equal results in equal filter-characteristics for all branches. If they were not the same, one would have to digitally correct for the differences to keep amplitude accuracy. Because correction factors are already required for the attenuation of all branches, one would need a two-dimensional correction table.<sup>1</sup>

The drain-source voltage  $V_{\rm DS}$  of an NMOS used as switch modulates the resistance of that switch, and hence reduces linearity [86]. One would prefer an ideal switch with zero resistance. However, making the switches bigger to lower the resistance increases parasitic capacitances, which reduces the system bandwidth (which should be 6 GHz). This limits the maximum size of the switches.

To reduce the body effect, the bulk connections of all switches are connected to the source/drain [87]. For the A-switches, the choice was made to connect it to the terminal connected to the same node as the load capacitor. This reduces the influence of the parasitic capacitance.

The *B*-switches are placed below the *b*-resistors, because if they were interchanged,  $V_{\text{GS}}$  would be lower and linearity would be reduced. The bulk is connected to ground, which as an additional advantage allows for easy layouting.

The bandwidth of the mixer directly depends on the RC-time. If the mixer is connected to branch n, the resistance  $R_C[n]$  seen from the capacitor needs to be calculated. To do this, it is necessary to find the impedance seen from node n (see fig. 5.3) looking up and looking down. Looking down is easy, because that is simply  $2R \Omega$ . The impedance  $Z_{up}[n]$ can be found in a recursive manner, similar to the calculation of  $Z_{in}[n]$ .

$$Z_{\rm up}[1] = Z_{\rm A} \ [\Omega]$$
$$Z_{\rm up}[n] = R + \frac{Z_{\rm A} Z_{\rm up}[n-1]}{Z_{\rm A} + Z_{\rm up}[n-1]} \ [\Omega]$$

from which the general expression can be derived (in a similar way as for  $Z_{in}[n]$ )

$$Z_{\rm up}[n] = 4R \frac{(4^n - 4)R + (4^n + 2)Z_{\rm A}}{(4^n + 8)R + (4^n - 4)Z_{\rm A}} \ [\Omega]$$

which quickly converges from  $Z_A \Omega$  to  $2R \Omega$ .

The total result then is

$$R_C[n] = R_{\rm on} + R_{n,a} + \frac{2RZ_{\rm up}[n]}{2R + Z_{\rm up}[n]} \ [\Omega]$$
(5.5)

which varies from  $R_{\rm on} + R_{1a} + Z_{\rm A}R/(R + Z_{\rm A})$  for n = 1 to  $R_{\rm on} + R_{n,a} + R$  for  $n \to \infty$ . With  $Z_{\rm A} = 50 \ \Omega$  and  $R = 35 \ \Omega$  the difference is 5.8  $\Omega$ , which can be corrected for by setting  $R_{1a} = R_{\infty,a} + 5.8 \ \Omega$ .

<sup>&</sup>lt;sup>1</sup>Note that another solution would be to keep the RC-product low enough such that any difference would still not have a significant influence on the desired IF-bandwidth. The downside then is that an integrated filter is effectively lost, requiring more filtering in the IF-part.



Figure 5.9: RF-frontend with component values used for simulation.

## 5.3 Simulation results

With the topology of fig. 5.8, the impedance seen from the loading capacitor ranges roughly from 30  $\Omega$  to 35  $\Omega$  in case the *a*-resistors are set to 0  $\Omega$ . To maximize linearity, one wants the switches to be as large as possible, while still adhering to the bandwidth specifications. Using ProMost, one can find that  $W/L = 100/0.06 \ \mu m$  yields an on-resistance of  $R_{\rm on} = 5 \ \Omega^{2}$  This means the resistance seen from the loading capacitor is 40  $\Omega$ . For a bandwidth of 100 MHz this results in C = 10 pF.

Simulations using these values show that the impedance seen from the capacitor changes about 10% when the frequency changes from 400 MHz to 6 GHz.<sup>3</sup> This difference is explained by the overlap capacitances of the switches, which makes the impedance seen from capacitor C frequency dependent. Reducing the dimensions of the switches to W/L = $50/0.06 \ \mu m$  reduces this problem to a 5% change in impedance over the whole frequency range, which is considered a better starting point. The on-resistance is doubled to 10  $\Omega$ , reducing the loading capacitance to C = 8.8 pF. The entire system is shown in fig. 5.9.

#### RC-bandwidth

The *RC*-bandwidth was determined using SpectreRF's PSS and PAC-analysis. Simulations of the *RC*-bandwidth for different attenuation settings showed that  $R_{1a}$  should be set to 5  $\Omega$ , and  $R_{2a}$  to 2  $\Omega$ , while  $R_{k,a}$ , k > 2, can be set to 0  $\Omega$ . The *RC*-bandwidths found in this case are shown in table 5.2, and they are clearly very close to the desired 100 MHz.

<sup>&</sup>lt;sup>2</sup>For the technology used, the on-resistance of a minimum-length NMOS ( $L = 0.06 \ \mu$ m), with  $V_{\rm GS} = 1.2$  V and  $V_{\rm DS} = 0$  V, can be approximated by 490/W  $\Omega$ , where W is given in  $\mu$ m. This equation is simply derived by observation and is also used in simulations when circuit parameters depend on the on-resistance of the switches.

<sup>&</sup>lt;sup>3</sup>400 MHz is chosen because of the constraint  $f_{-3dB} \ll f_{LO}$ , see eq. (5.4).

Table 5.2: RC-bandwidth found for different attenuation settings, where 100 MHz is the desired value.

$f_{\rm LO}$	I	BW [M	Hz] usin	g brancl	1
[GHz]	1	2	3	4	5
0.4	99.0	96.4	98.5	97.7	97.4
6.0	101.7	99.6	101.7	101.2	100.7

Table 5.3: CG found for different attenuation settings and the pre-calculated values.

$f_{\rm LO}$		CG [	dB] using	g branch	
[GHz]	1	2	3	4	5
0.4	0.42	-5.26	-11.25	-17.37	-23.63
6.0	0.26	-5.45	-11.45	-17.58	-23.84
Calc.	0.41	-5.24	-11.17	-17.16	-23.18



Figure 5.10: Simulation results for the  $S_{11}$ -parameter.

#### Conversion gain

The conversion gain was determined using the same simulation as for determining the RC-bandwidth. The simulated CG of the entire system is shown in table 5.3. It is very close to the expected CG shown in table 5.1 (and repeated in table 5.3 for convenience). The lower CG at 6 GHz is caused by the Bandwidth (BW) of the system.

## Impedance matching

The scattering parameter  $S_{11}$  was simulated using SpectreRF's QPSP-analysis and is shown in fig. 5.10. The value varies somewhat over the whole frequency range, but always stays below -14.9 dB for any attenuator setting, which is clearly better than the desired -10 dB, and very close to the theoretically calculated -15.1 dB. One can also see that  $S_{11}$  starts to deteriorate fast near 6 GHz, which indicates the BW of the system is not much higher than 6 GHz.



Figure 5.11: Simulation results for the NF at an offset of 1 MHz.

#### Noise Figure

The NF was determined using SpectreRF's PNOISE-analysis. The simulated NF (with the input frequency equal to the oscillator frequency and looking at a 1 MHz offset) is shown in fig. 5.11.

The NF is around 11.2 dB for branch 1 at higher frequencies, which is equal to the expected 11.2 dB (see section B.10 for the derivation). For the other branches, the NF goes up in steps of about 6 dB (17.1, 23.1, 29.2 and 35.5 dB respectively), which is as expected as the signal power is decreased in steps of approximately 6 dB, while the amount of noise remains roughly the same.

The NF clearly goes up for lower frequencies. In CMOS, flicker noise usually causes the NF to go up at low frequencies, because it has a power spectrum inversely proportional with frequency. At some point, known as the *corner frequency*  $f_C$ , flicker noise drops below the thermal noise. Because  $V_{\rm DS}$  is very low and the transistor is in the triode region,  $f_C$ is expected to be rather low. Simulations show  $f_C$  to be in or below the kHz-region (an example for  $f_{\rm LO} = f_{\rm in} = 300$  MHz is shown in fig. 5.12), so flicker noise does not explain the increase in NF for frequencies below 1 GHz.

Using SpectreRF's Noise Summary, the switch noise contribution for any branch is about 25% at 3 GHz. The flicker noise contribution indeed is insignificant. At 100 MHz, the relative noise contributions of all the components are the same as at 3 GHz, only all of them are about twice as large. This corresponds with the increase in NF of about 3 dB.

It is conjectured, that the cause of the increase of the noise is the transfer function of the Tayloe mixer. Equation (5.4) only holds for  $f_{-3dB} \ll f_{LO}$ , and this is precisely violated in the region where the NF increases. At lower frequencies, the voltage on the capacitor



Figure 5.12: NF as a function of offset-frequency for  $f_{\rm LO} = f_{\rm in} = 300$  MHz.

Table 5.4: IP3 found for different attenua-tion settings

$f_{\rm LO}$	IF	P3 [dBr	n] usin	g brand	ch
[GHz]	1	2	3	4	5
0.4	25.6	32.0	38.0	39.5	38.6
6.0	23.7	29.9	32.5	38.0	36.4

will be much better at following the input voltage, thereby reducing the filtering effect. The Tayloe mixer starts to act more like a sampling mixer, which has a higher NF. This would also explain why the relative noise contributions of the various components do not change.

Because the simulated circuit is unbalanced and does not use I/Q-mixing, it is expected that the NF can be brought down by 6 dB if both methods are included. However, both balancing and I/Q-mixing require extra switching transistors, which may influence linearity and BW because of parasitic capacitances, and careful simulations should show whether requirements can still be met (see recommendations).

# Linearity

Linearity was determined using SpectreRF's QPSS-analysis. The simulated IP3 is shown in table 5.4 for  $f_{\rm LO} = 400$  MHz with two input tones of 410 MHz and 411 MHz and for  $f_{\rm LO} = 6.0$  GHz with two input tones of 6.010 GHz and 6.011 GHz. An example is shown in fig. 5.13.

For minimal attenuation, IP3 is roughly +25 dBm at low frequencies. This is very close to the value found by Soer [60, fig. 6.12], which is to be expected as the circuits are very similar except for the matching network.

A more detailed IP3 as a function of frequency is shown in fig. 5.14. An important observation is that IP3 goes down for higher frequencies. This is caused by the oscillator buffer, which is implemented in the simulation as an inverter, consisting of a PMOS with dimensions  $W/L = 100/0.06 \ \mu\text{m}$  and an NMOS with dimensions of  $W/L = 45/0.06 \ \mu\text{m}$ . In the simulations it is being driven by a near-ideal oscillator with a rise and fall time of 1 fs.<sup>4</sup> The relative size PMOS:NMOS=20:9 gives an equal rise and fall time. The oscillator buffer is depicted in fig. 5.15.

 $<sup>^4\</sup>mathrm{In}$  retrospect it would have been better to give the oscillator a rise and fall time realistic to 65 nm CMOS to better resemble a real oscillator.



Figure 5.13: Simulation results for IP3 using branch 1 at  $f_{\rm LO}=400$  MHz.



Figure 5.14: IP3 as a function of frequency.



Figure 5.15: The oscillator buffer is an inverter scaled for equal rise and fall time, and is driven by a near-ideal pulse source.



Figure 5.16: Simulation results for IP3 using branch 1 at 6 GHz for different oscillator buffer sizes with ratio PMOS:NMOS=20:9.

Table 5.5: IP3 at  $f_{\rm LO} = 400$  MHz found for branch 5 when all switches, except for  $A_5$  and one other switch (shown in the table), are replaced by ideal short-circuits and open connections.

		IP3 [c	dBm] u	sing sv	witch $A$	$_5$ and		
$A_1$	$A_2$	$A_3$	$A_4$	$B_1$	$B_2$	$B_3$	$B_4$	$B_5$
35.2	43.9	49.9	51.2	36.7	42.9	49.2	51.5	51.3

At 6 GHz and a duty cycle of 25%, the time the signal is high is about 42 ps. With a rise and fall time in the order of 10 to 20 ps, the duty cycle can be heavily influenced as the switch turns on somewhere near 400 mV. Moreover,  $V_{GS}$  is less than 1.2 V and is changing most of the time, which changes the on-resistance and thus introduces a nonlinear element in the circuit. A larger inverter can change the gate-voltage of the switches faster, reducing this nonlinear effect. The size of the inverter was swept to determine the smallest size for which linearity is not significantly affected. The results are shown in fig. 5.16.

Apparently, the oscillator frequency of 6 GHz is simply too high to obtain a linearity comparable to that at lower frequencies for this 65 nm technology. With a PMOS-size of 100 mm (not shown), IP3 at 6 GHz is +24.7 dBm, only a few tenths of a dB better than at 200  $\mu$ m. Even without loading, the rise and fall time of the inverter is about 10 ps. Some extra tweaking with the relative sizes of the PMOS and NMOS may improve IP3 a little bit more, but not much. Because of the small improvement in IP3 by greatly increasing the size of the inverter, the buffer size is kept at  $W = 100 \ \mu$ m for the PMOS and  $W = 45 \ \mu$ m for the NMOS, just as in the previous simulations.

Going back to table 5.4, one would expect IP3 to increase by the same amount as the input signal is attenuated. It can be seen from the simulation results that this is the case for branches 2 and 3, but not for branches 4 and 5.

Since the only nonlinear elements in the circuit are the switches, it was conjectured that the switches limit linearity, even when they are fully on  $(V_{GS} = 1.2 \text{ V})$  or fully off  $(V_{GS} = 0 \text{ V})$ . Simulations were performed using branch 5 and switch  $A_5$ . When all other switches are replaced by an ideal short or open connection, IP3 was simulated to be 51.5 dBm, roughly  $4 \cdot 6 = 24$  dB better than using branch 1, which is to be expected as the signal is attenuated by an extra 24 dB. Replacing all other switches except one by an ideal short or open connection gave the IP3 values shown in table 5.5.

Although all switches are limiting linearity, the switches in the first branch limit it most.



Figure 5.17: Simulation results for IP3 for different W at  $f_{\rm LO} = 400$  MHz with  $W_{\rm PMOS} = 100 \ \mu {\rm m}$ .



Figure 5.18: Simulation results for IP3 for different  $R_C$  at  $f_{\rm LO} = 400$  MHz.

This is not surprising as they receive the largest signal swings.

Since non-linearity is present in both the resistance and the capacitance of the switches, it is expected that linearity can be optimized by changing W. The complete circuit was simulated for different W with the attenuation set to branch 1. Because a different Wchanges the on-resistance of the switches, C was changed as well to keep the RC-bandwidth at 100 MHz. For example, with  $W = 80 \ \mu m$ ,  $R_{\rm on} \approx 6 \ \Omega$ , so  $R_C \approx 41 \ \Omega$  and  $C = 9.7 \ {\rm pF}$ . The results are shown in fig. 5.17 and do not change if  $W_{\rm PMOS}$  is increased to 1000  $\mu m$ , which means the size of the oscillator buffer is not limiting linearity here. From these results it can be concluded that  $W = 50 \ \mu m$  gives a linearity only 1 dB from the optimum value, while still adhering to the 6 GHz BW of the entire system.

#### RC-product

A degree of freedom in the circuit is the relative value of R and C in the RC-product. Using  $W = 50 \ \mu m$ ,  $R_C$  was swept by changing the *a*-resistors (see figs. 5.6 and 5.8). Because the *a*-resistors cannot have a resistance larger than 60  $\Omega$  due to the on-resistance of the switches and the impedance matching requirement, and  $R_{1,a}$  is already 5  $\Omega$  to start with,  $R_C$  can be swept from 45  $\Omega$  to 100  $\Omega$ . C is changed accordingly to keep RC at the 100 MHz bandwidth. The results are shown in fig. 5.18

From these results one would conclude that increasing  $R_C$  is beneficial. Inspection of the circuit reveals that increasing the *a*-resistors increases the noise at the output, increasing the NF. The NF has also been simulated, and is shown in fig. 5.19a. For the SFDR only the value IP3 minus NF is important, and can therefore be thought of as some sort of FoM



Figure 5.19: Simulation results for NF and the FoM (IP3-NF) for different  $R_C$  at  $f_{\rm LO} = 400$  MHz.

Table 5.6: 1-dB CP found for different attenuation settings at  $f_{\rm LO} = 400$  MHz.

1-d	B CP [	dBm] u	sing bra	nch
1	2	3	4	5
-3.6	+0.2	+2.1	+4.0	+8.6

(note that, unlike the SFDR, this FoM is independent of RBW, and hence provides a good measure for comparison). It is shown in fig. 5.19b. Note that correlation cannot reduce the noise here, because this is a single attenuator. The attenuator for a correlation SA will be discussed in ??.

#### Measurement range

The measurement range is the range of signals that can be measured, but not necessarily at the same time.

The largest signal can be defined as the signal that is just below blowing up the circuit, but the choice made here is the 1-dB Compression Point (CP) of the circuit. The reason for this is, that at the 1-dB CP the gain error becomes 1 dB and harmonic content is introduced, making the resulting spectrum unreliable. The 1-dB CP simulation results are shown in table 5.6. Clearly the 1-dB CP goes up for larger attenuation, with the maximum at +8.6 dBm. This maximum corresponds (at 50  $\Omega$ ) to a sinusoid with an amplitude of 850 mV, so the limit is simply set by the supply voltage.

The smallest signal that can be measured can be defined as being equal to the noise floor level. The minimal NF is 11.2 dB. To determine the total NF of the system, Friis' equation [44] can be used

$$F_{\text{total}} = F_1 + \frac{F_2 - 1}{G_1} + \frac{F_3 - 1}{G_1 G_2} + \dots + \frac{F_n - 1}{\prod_{k=1}^{n-1} G_k}$$
(5.6)

where  $F_i$  denotes the noise factor and  $G_i$  the available power gain of the *i*-th stage. Although this formula requires the noise factor of a stage to be tailored for the output impedance of



Figure 5.20: Schematic of the RF-frontend for a correlation SA using two paths. This example has only three branches.

the stage driving it, it can be (ab)used to get an estimate on the influence of the IF-part of the system.

Since IF-circuits are generally easier to make than RF-circuits, a reasonable assumption is that the IF-part has an NF of 5 dB. With an NF of 11.2 dB and a CG of 0.4 dB of the RF-part, the total NF becomes 11.8 dB.

The sensitivity of an SA goes up if the RBW is lowered [3], which is also logical if one takes in mind that the power of a sinusoid (or any other signal that is narrowband enough for that matter) in that BW remains the same, while the noise power scales with the RBW. Although RBW-specifications are not used in this frontend, the specifications from [4] will be used, where a minimum RBW of 10 kHz is specified. The noise power in a 10 kHz bandwidth is  $kTB = 4 \times 10^{-17}$  W, so with an NF of 11.8 dB (F = 15.1) the equivalent input-referred noise power is  $kTBF = 6.1 \times 10^{-16}$  W. The smallest signal that can be detected is equal to this noise level: -122 dBm.

#### Power consumption

The entire simulated circuit is passive, except for the oscillator. Because an ideal oscillator is used, the only part consuming power is the oscillator buffer. This was simulated using transient analysis and integrating the current through the source of the PMOS. The energy per cycle is 0.42 pJ, so at 6 GHz the power consumption is roughly 2.5 mW It is assumed that the static power consumption of the buffer is negligible, such that its power consumption scales linearly with the oscillator frequency.

#### 5.4 Two attenuators

In crosscorrelation, two paths are required. The earlier the input signal is split, the better, because all noise added before the split will be fully correlated and therefore cannot be correlated away. In this section the frontend is redesigned with two identical attenuators in parallel, both directly connected to the antenna, as shown in fig. 5.20.

#### 5.4.1 Circuit design

Two attenuators in parallel means that R needs to be doubled to 70  $\Omega$  to achieve impedance matching. With two attenuators, one needs to recalculate all important parameters.

#### 5.4. TWO ATTENUATORS

Per attenuator the input impedance still is

$$Z_{\rm in}[n] = \frac{4^n + 2}{4^n - 1} R \ [\Omega]$$

The total input impedance then is

$$Z_{\rm tot}[n] = \frac{1}{2} \frac{4^n + 2}{4^n - 1} R \ [\Omega] \tag{5.7}$$

For the voltage at a node one finds

$$V_{1,n} = \frac{Z_{\text{tot}}[n]}{Z_{\text{tot}}[n] + Z_{\text{A}}} V_{\text{in}} \quad [V]$$
$$V_{k,n} = \frac{Z_{\text{in}}[n-k+1]}{Z_{\text{in}}[n-k+1] + R} V_{k-1,n} \quad [V]$$

from which can be calculated that with  $R = 70 \Omega$ , the attenuation is equal to the situation with one attenuator and  $R = 35 \Omega$ .

To determine the RC-bandwidth, it is important to know whether one or two switches are open at the same time. If two switches are open, the capacitors 'see' each other. This will influence the impedance seen from the capacitors, and thus change the RC-bandwidth.

Consider the input impedance of a single attenuator when switch  $A_1$  is open. It can then be observed that this is equal to  $2R \Omega$  in parallel with  $R_{\text{on}} + R_{1,a} + 1/j\omega C$ . Now, with  $R = 70 \ \Omega$ ,  $R_{\text{on}} = 10 \ \Omega$ ,  $R_{1,a} = 41 \ \Omega$  and C = 5.0 pF (these values will be derived later), the impedance is 140  $\Omega$  at DC and only 51  $\Omega$  at 6 GHz. Although this difference is less for branches with a larger attenuation, because the capacitors 'see' each other to a lesser extent, it still means that the *RC*-bandwidth depends on the input frequency.

As discussed before, making the RC-bandwidth large enough to cope with the fluctuations may be a solution. The solution presented here is to pass the signal in one branch when it is not passed in the other. With a duty cycle of 25%, the oscillator phase in one path can be shifted anywhere between 90° and 180° with respect to the oscillator phase in the other path. This phase difference gives a phase difference between the two paths, but that can be corrected for in the digital domain.

A disadvantage is that for I/Q systems and balanced systems, one also requires 90° or 180° phase shifts of the oscillator. Because four different oscillator phases occupy all available time, neither of these techniques can be used if the oscillator has a phase shift of something other than 90° or 180°. Only one of them can be used if the phase shift is 90° or 180°. An important advantage of having one switch open at a time is that all thermal noise added by the circuit will be completely uncorrelated in the branches. Flicker noise changes only slowly, so there is still some correlation among the branches. Because flicker noise concentrates around DC and  $f_C$  is so low, this will most likely not be a big practical problem. It will however introduce a limit on noise reduction.

An important observation is that in case no correlation is required, both paths can be used as a balanced pair, decreasing NF by 3 dB. This requires twice the amount of energy, but reducing NF by 3 dB using correlation requires four times the amount of energy.

Using the fact that only one switch is open at the same time, one can calculate the impedance looking up from one voltage node. Defining  $Z_{up}[k, n]$  as the impedance seen looking up from node  $V_{k,n}$ , one finds

$$Z_{\rm up}[1,n] = \frac{Z_{\rm A} Z_{\rm in}[n]}{Z_{\rm A} + Z_{\rm in}[n]} \ [\Omega]$$
$$Z_{\rm up}[k,n] = R + \frac{2R Z_{\rm up}[k-1,n]}{2R + Z_{\rm up}[k-1,n]} \ [\Omega]$$

The total result then is

$$R_C[n] = R_{\rm on} + R_{n,a} + \frac{2RZ_{\rm up}[n,n]}{2R + Z_{\rm up}[n,n]} \ [\Omega]$$
(5.8)



Figure 5.21: RF-frontend of a correlation SA with component values used for simulation. Only one of the two attenuators is shown as they are equal.

which varies from  $R_{\rm on} + R_{1a} + Z_{\rm A}R/(R + Z_{\rm A})$  for n = 1 to  $R_{\rm on} + R_{n,a} + R$  for  $n \to \infty$ . With  $Z_{\rm A} = 50 \ \Omega$  and  $R = 70 \ \Omega$  the maximum difference is 41  $\Omega$ , which can again be corrected for by properly setting the *a*-resistors.

The results found through simulation are  $R_{1,a} = 41 \ \Omega$ ,  $R_{2,a} = 13 \ \Omega$ ,  $R_{3,a} = 4 \ \Omega$ ,  $R_{4,a} = 1 \ \Omega$  and  $R_{5,a} = 0 \ \Omega$ . With  $W/L = 50/0.06 \ \mu \text{m}$  of the switches,  $R_{\text{on}} = 10 \ \Omega$ , and hence for a 100 MHz bandwidth C = 5.0 pF.

One attenuator of the resulting circuit is shown in fig. 5.21. The  $S_{11}$ -parameters, attenuation, *RC*-bandwidth and IP3 are expected to be very comparable to the case with one attenuator. NF is expected to increase due to larger resistors in series with the mixing switch. In fact, in an *RC*-circuit it can be calculated that the noise power on a capacitor is equal to kT/C [27], and since *C* has changed from 8.8 pF to 5.0 pF, an increase in NF of  $10 \log_{10}(8.8/5.0) = 2.5$  dB is expected.

#### 5.4.2 Simulation results

All simulations are identical to the simulations with one attenuator. Some results are shown in table 5.7.

The CL changes 0.7 dB over the whole bandwidth and is slightly higher than the case of one attenuator. The *RC*-bandwidth spreads more due to the increase in number of switches, which increases parasitic capacitances. It is comparable to the simulation results with one attenuator and switches with  $W = 100 \ \mu m$ . It is still pretty close to the desired 100 MHz though.

While IP3 at 400 MHz is slightly higher, IP3 at 6 GHz is somewhat lower than with one attenuator. It is not clear what causes this difference.

All other values are comparable to the situation with one attenuator, except the NF, which is 2.5 dB higher, as expected. Note for example the NF of 14.1 dB for  $R_C = 80 \Omega$  in fig. 5.19a, which is exactly equal to the 14.1 dB found here. It does not make any significant difference for CL, BW and NF whether the oscillator phase is 90° or 180° shifted. For IP3

	f	$\Delta \phi$			Branch		
	$[\mathrm{GHz}]$	[°]	1	2	3	4	5
	0.4	90	0.39	-5.30	-11.32	-17.50	-22.89
CC [4P]	0.4	180	0.40	-5.30	-11.32	-17.50	-22.90
CG [db]	6.0	90	-0.32	-5.91	-11.93	-18.13	-23.50
	6.0	180	-0.26	-5.89	-11.93	-18.13	-23.50
	0.4	90	95.4	96.2	95.7	96.1	96.0
	0.4	180	95.0	96.3	95.7	96.0	96.2
DW [MHZ]	6.0	90	106.2	106.0	106.0	106.5	105.7
	6.0	180	105.5	105.6	105.8	106.5	105.6
	0.4	90	26.1	32.0	36.8	37.9	35.5
ID9 [dDma]	0.4	180	26.5	32.1	36.8	38.1	35.4
IP3 [dDm]	6.0	90	21.1	25.6	31.1	34.1	33.4
	6.0	180	22.0	25.7	31.1	34.0	33.4
	0.4	90	14.1	19.8	25.9	32.0	38.5
NE [JD]	0.4	180	14.1	19.8	25.9	32.0	38.5
INF [UD]	6.0	90	14.2	19.8	25.8	32.0	38.4
	6.0	180	14.2	19.7	25.8	32.0	38.4

Table 5.7: Simulation results with two attenuators in parallel. NF is determined at 1 MHz offset.



Figure 5.22: Simulation results for  $S_{11}$ -parameter with two attenuators in parallel.

the difference is a few dB, but since the IF-circuitry will most likely limit linearity somewhat more, it is not expected to have a significant impact on overall system linearity. This means that in a design one still has the freedom to choose between balancing and I/Q-mixing.

Simulation results for impedance matching are shown in fig. 5.22. It is only slightly worse than with one attenuator, but still  $S_{11} < -14.5$  dB in all cases over the whole frequency range.

IP3 as a function of frequency is shown in fig. 5.23. Again a slight degradation for higher frequencies can be observed, just as in the case with one attenuator. An important observation is that the degradation is higher for a  $90^{\circ}$  phaseshift. This can be explained by the fact that the higher the frequency, the more relative overlap there is between the two oscillator phases, due to the finite transition time of the switches.

The effect of increasing the oscillator buffer to mitigate the loss in linearity at higher frequencies is again simulated. The results are shown in fig. 5.24. Linearity at 6 GHz is



Figure 5.23: Simulation results for IP3 as a function of frequency with two attenuators in parallel.



Figure 5.24: Simulation results for IP3 as a function of oscillator buffer size (PMOS:NMOS=20:9) with two attenuators in parallel.



Figure 5.25: Simulation results for IP3 at 400 MHz as a function of W with two attenuators in parallel and 180° phase difference.

obviously lower than at 400 MHz. Increasing  $W_{\rm PMOS}$  to 1000  $\mu$ m (not shown) increases IP3 to +22.4 dBm at 90° and +22.8 dBm at 180°, so only a little more can be gained at the expense of a lot of power. At 400 MHz increasing  $W_{\rm PMOS}$  to more than 200  $\mu$ m does not increase linearity.

Changing the width of the switches results in the graph shown in fig. 5.25. The chosen  $W = 50 \ \mu \text{m}$  is about the optimum in terms of linearity.

Just as with the single attenuator case, the impedance seen from the capacitor can be



(b) NF as a function of  $R_C$  at 400 MHz  $\,$  (c) FoM as a function of  $R_C$  at 400 MHz with  $W=50~\mu{\rm m}$   $\,$  with  $W=50~\mu{\rm m}$ 

Figure 5.26: Simulation results for IP3, NF and the FoM (IP3-NF) for different  $R_C$  at  $f_{\rm LO} = 400$  MHz with two attenuators in parallel.

Table 5.8: 1-dB CP found for different attenuation settings at  $f_{\rm LO} = 400$  MHz.

1-d	IB CP [	dBm] u	sing bra	nch
1	2	3	4	5
-3.3	+1.1	+3.1	+3.9	+7.7

varied. Looking at fig. 5.26, increasing  $R_C$  shows a decrease in IP3 and an increase in NF, so keeping  $R_C$  at the lowest possible value, 80  $\Omega$ , is the best option. The FoM is 1.5 dB less than in the case where only one attenuator is present in the system, but with correlation this 1.5 dB can be gained at the cost of a twofold increase in measurement time.

For determining measurement range, the 1-dB CP is determined for all attenuation settings, see table 5.8. The largest signal that can be detected is +7.7 dBm, roughly equal to the single attenuator. The smallest signal that can be detected is a signal with a power equal to the residual correlated noise, which at this moment is still unknown, but it is certainly smaller than in the case of a single attenuator.

The power consumption is now simply twice that of a single attenuator, because two oscillator buffers are required, so it is 5.0 mW.

# 5.5 SFDR of analog frontend

The goal of the analog frontend design is to make it as linear as possible to maximize the SFDR, as the noise can be reduced by correlation. Because correlation may not always be an option, for instance due to limited available measurement time, a first estimate of the SFDR can be given by using the simulated results on noise and linearity of the single attenuator. It is assumed that third-order modulation is the limiting factor, and not second-order modulation or any other type of non-linearity.

The estimation formula for the SFDR is [19]

SFDR = 
$$\frac{2}{3}$$
 (IP<sub>3</sub> - L<sub>N</sub>) =  $\frac{2}{3}$  (IP<sub>3</sub> + 174 - 10 log B - NF) [dB] (5.9)

where  $L_N$  is the noise level, consisting of the thermal noise floor of kT = -174 dBm/Hz in combination with the NF, integrated over the bandwidth *B*. It can be used to estimate the maximum SFDR this frontend can achieve. The NF is assumed to be 11.8 dB as derived in section 5.3. With B = 1 MHz, the SFDR of this frontend can be 84 dB at low frequencies, and 82 dB at 6 GHz, both far beyond the goal of 70 dB.

Using this information, the required number of bits of the ADC can be determined to also achieve 84 dB of SFDR. It is assumed that 100 MHz of bandwidth is passed through the ADC and the input signal only contains thermal noise. Worst case, the total input noise power equals -174+80 = -94 dBm, or  $4.0 \times 10^{-13}$  W. With a NF of 11.8 dB, the total noise power equals -94 + 11.8 = -82.2 dBm, or  $6.0 \times 10^{-12}$  W. Assuming the maximum input amplitude of a sine wave is 0.5 V, its power (at 50  $\Omega$ ) is  $5 \times 10^{-3}$  W. The SNR then is 89 dB. Using the results from chapter 3, the required number of bits for an SFDR of 84 dB is 9.7 bits. With 9.7 bits however, the SNR of the ADC is only 60 dB (see eq. (3.3)), which will increase the NF of the system at the minimum attenuation setting by a staggering 29 dB. A 15-bit ADC has an SNR of 92 dB and would therefore not increase NF significantly.

# 5.6 Comparison with other spectrum analyzers

To put the derived simulation results in perspective, table 5.9 compares the figures obtained using the current circuit to the figures obtained from several commercial SAs, as well as some SAs from literature. This comparison is only preliminary, as many other factors not taken into account here may negatively influence the results obtained in this thesis. Moreover, other important aspects such as measurement time, phase noise, amplitude accuracy and IP2 are not even considered here. Nevertheless, it gives a clue as to some of the limitations and opportunities of an integrated SA using this RF-frontend.

The Voltage Standing Wave Ratio (VSWR) is another measure to indicate impedance matching, and is related to the  $S_{11}$ -parameter by (note that it is customary to define  $S_{11}$  in dB by using  $20 \log S_{11}$ , which is also what has been done in this thesis)

$$VSWR = \frac{1+S_{11}}{1-S_{11}}$$
(5.10)

This gives the current implementation a VSWR of less than 1.5, and therefore quite comparable to the impedance matching of the commercial SAs. In case the 25% spread in R is incorporated, a VSWR of less than 1.8 is obtained.

The most significant difference is the NF, which is 7 dB better than the next best in the list. A low NF implies a larger measurement range and a higher SFDR, which can also be observed from the table. It must be noted that phase noise will lower the SFDR (from the data presented in [4] by about 7 dB), which makes the difference in SFDR smaller.

Linearity of the designed frontend is among the best in the list, but as stated previously, the IF-part will be very hard to design with such high linearity.

The 1-dB CP was chosen as the upperbound on measurement range, while the commercial SAs use a different definition. Because 65 nm CMOS has a supply voltage of 1.2 V, which is

Table 5.9	): Comparison	of per:	formance fig	ures for the de	signed fi	rontend	with other	SAs.	
Spectrum Analyzer	Type	Price [k€]	Freq. range [GHz] (rounded)	Meas. range [dBm] (1 MHz RBW)	VSWR	NF [dB]	IP3 [dBm] (min. att.)	1-dB CP [dBm] (min. att.)	SFDR [dB] (1 MHz RBW)
This design (single att.)	on-chip		9-0	-102/+8	< 1.8	12	25	-4	84
This design (dual att.)	on-chip		0-0	< -99/+8	< 1.8	< 15	24	ဂု	> 82
Hameg HM5014-2 [88]	stand-alone	ŝ	$^{0-1}$	-80/+10	< 1.5	34	4	ż	51
Tektronix RSA2203A [4, 89]	stand-alone	21	0-3	-80/+30	< 1.4	24	30	0	20
Tektronix RSA2208A [4, 89]	stand-alone	ċ	$^{0-8}$	-80/+30	< 1.4	24	30	0	20
Rohde & Schwarz FSP 3G [4, 90]	stand-alone	17	0-3	-90/+20	< 2.0	22	10	0	58
Rohde & Schwarz FSP 7G [4, 90]	stand-alone	ċ	2-0	-90/+20	< 2.0	22	10	0	58
Agilent PSA E4443A [4, 91]	stand-alone	43	2-0	-91/+30	< 1.6	19	16	ŝ	20
Agilent N9340A [92]	stand-alone	9	$0^{-3}$	-99/+33	< 1.8	34	10	ż	< 60
SED Decimator PCI [93]	PC peripheral	ŝ	1-2	-75/-5	\$	\$	ż	ż	2
Metrix MTX1050 PC USB SA [94]	PC peripheral	1	0-1	-71/+20	ن	53	0	ż	< 40
[95]	PCB		$^{0-1}$	-98/?	ż	$19/37^{a}$	ż	ż	20
[10]	on-chip		$0^{-3}$	2/2	< 4.4	ب	\$	ż	2
[11]	on-chip		$0 - 10^{-5}$	2/2	ć	ż	0	ż	2
[12]	on-chip		$0 - 10^{-3}$	i/i	\$	ż	ż	ċ	;
<sup>a</sup> Given in paper / calculated usin	ng -99 dBm DAN	IL at 71	kHz RBW						

<sup>g</sup> Not available

also the maximum constant voltage over the gate-oxide, the highest amplitude at the input of a transistor is 0.6 V, which at 50  $\Omega$  is  $\frac{1}{2}A^2/50 = 3.6$  mW, or +5.6 dBm. This means that in this specific case the 1-dB CP also roughly equals the maximum input power, which is definitely less than the other SAs. To obtain the same upper bound on measurement range as the commercial SAs, external attenuators are required.

# 5.7 Conclusions

The RF-part of an analog frontend for an SA was designed to maximize linearity. It has impedance matching over the entire range of 0 GHz to 6 GHz, with a selectable attenuation with steps of roughly 6 dB, starting at a gain of 1.3 dB. The system uses a Tayloe mixer for frequency conversion with an RC-bandwidth of 100 MHz. The Tayloe mixer introduces 0.9 dB loss. The idea is that gain will be provided in the IF-part, because then it can be done at lower frequencies and therefore using less power and achieving higher linearity.

Two implementations of the RF-part were designed, one for a regular SA (with only one path), and one for a correlation SA, which has two paths, each containing an attenuator. The only active part is the oscillator buffer. With the chosen topology, each oscillator buffer consumes 2.5 mW at 6 GHz, which means a power consumption of 5 mW for the variant with two attenuators.

With a single attenuator, the NF is 11.2 dB at minimum attenuation. IP3 varies from +25.6 dBm at 400 MHz to +23.7 dBm at 6 GHz. At low frequencies linearity is limited by nonlinearity in the NMOS-switches, while at high frequencies it is limited by the oscillator buffer. Both limitations are expected to improve by newer technologies, as smaller features reduce on-resistance, parasitic capacitance and switching speed of an inverter. IP3 saturates at roughly +38 dBm for larger attenuation, which limits the useful amount of attenuation steps.

For the pair of attenuators, the NF is 2.5 dB higher. To prevent any problems with respect to the RC-bandwidth of the first-order filter inherent in a Tayloe mixer, the switches in both attenuators cannot be on at the same time (see the recommendations). The oscillator signal in the two attenuators can have a phase difference of 90° or 180°. Performance figures are roughly the same for both options: IP3 varies from +26.3 dBm at 400 MHz to +21.5 dBm at 6 GHz.

Comparing to commercial SAs, the IP3 of this design is among the best, the NF is much better, impedance matching is comparable and the 1-dB CP is worse. The latter is caused by the limited voltage supply of 65 nm CMOS, which also limits the maximum input power to levels more than 20 dB lower than that of commercial SAs. The low NF, which can be made even lower with correlation, allows much smaller signals to be detected, and allows a larger SFDR.

Using the assumption that the NF at IF is 5 dB and linearity is not limited by the IF-circuitry, the SFDR of this frontend can be 82 dB to 84 dB (with an RBW of 1 MHz) without correlation, far more than the required 70 dB. With correlation, and under the assumption that there is no correlation between the noise realizations in both paths, the SFDR could be increased to 92 dB (the situation where the effective NF has been brought down to 0 dB), provided enough measurement time is available. The ADC should have at least 9.7 bits to achieve the same SFDR, but this will increase the NF significantly, for branch 1 by a staggering 29 dB. A 15-bit ADC does not significantly increase the NF and may therefore be a better option.

# 5.8 Recommendations

The current implementation works, but there are still certain aspects that can be worked out in more detail or that require some attention.

# 5.8.1 Increasing linearity of RF-frontend

The IF-part is neglected in the discussion. Using the approximate formula for cascaded stages [44, p.24 eq. (2.47)]

$$\frac{1}{\mathrm{IP}_{3_{\mathrm{total}}}^{2}} \approx \frac{1}{\mathrm{IP}_{3_{1}}^{2}} + \frac{G_{1}}{\mathrm{IP}_{3_{2}}^{2}} + \frac{\prod_{i=1}^{n-1} G_{i}}{\mathrm{IP}_{3_{n}}^{2}}$$
(5.11)

it can be calculated that with an IP3 of +26 dBm and a CG of 0.4 dB, the IF-circuitry needs to have a linearity of at least +20.3 dBm to accommodate a system linearity of +20 dBm. This is by no means an easy problem, especially because for proper AD-conversion of small input signals an amplifier is needed. Although the IF-part will most likely limit system linearity, there are still some possibilities to increase linearity of the RF-frontend, which may or may not be useful.

In simulations it was observed that the first switches limit linearity, such that the practical number of attenuation steps is limited. One solution may be to use a CMOS-switch (also known as a transmission gate) such that the switch resistance is more constant over a larger voltage range, and hence more linear [86].

Another solution might be to use bootstrapping to keep the gate-source voltage at a constant value [86]. A practical problem to bootstrapping is that it only works well with switching transistors. This is because bootstrapping circuits usually use a capacitor to keep the voltage difference constant. The capacitor voltage will slowly decrease and needs to be reset every once in a while. In the discussed circuit the switches that are constantly on also limit linearity, and it may be very hard to design a bootstrapping circuit for them.

In combination with the use of CMOS-switches and/or bootstrapping, the number of attenuation steps may be optimized such that there is more design room in choosing W of the switches before the system bandwidth drops below 6 GHz.

The duty cycle was chosen to be 25% because a Tayloe mixer has a good NF and CG. In this particular situation maybe a different duty cycle can help to improve linearity.

# 5.8.2 Increasing the measurement range

The upperbound on the measurement range is dictated by the supply voltage. External attenuators could be used to improve it, but another solution may be to use on-chip resistors on thick oxide before entering the attenuator, see fig. 5.27. This would require separate pins on the chip-interface because internal switches cannot be used at those input voltages. An important question is how far the upper limit on input power can be stretched, and what is required to retain impedance matching.

#### 5.8.3 Towards a real implementation

The current implementation serves mainly as a proof-of-concept. To make this easier, an unbalanced system is designed. In reality, one would want to be less sensitive to crosstalk and power and ground bounce, as these can cause distortion, or, if they can be regarded as noise, introduce correlated noise in the branches, which cannot be correlated away. Therefore, the design of a balanced circuit is recommended. An I/Q-implementation would also come in handy as some signals may be converted to DC.

PM or HR-mixing was suggested in chapter 4 as a solution to remove unwanted images and harmonics caused by the blockwave nature of the oscillator. An important problem is how this can be implemented. As discussed in section 5.4, multiple attenuator/mixer circuits having mixer-switches open at the same time gives a frequency-dependent bandwidth of the RC-filter in the Tayloe mixer. The current design uses a constant RC-bandwidth, but given the fact that filters at IF are required anyway, it may also be a solution to make the



Figure 5.27: On-chip resistors may be used to stretch the upper limit of the measurement range beyond the supply voltage.

RC-bandwidth wide enough such that any change in bandwidth does not filter the desired bandwidth. Because the resistance cannot be made smaller due to the impedance matching requirement, the capacitor needs to be smaller. This will have a detrimental effect on the NF, as kT/C increases.

Chapter 6

# **Digital Backend**

After the signal is downconverted to IF and filtered, it will be digitized by the ADC. The samples then need to be processed to obtain the spectrum.

This chapter starts with an overview of the possible implementations of a digital correlator.<sup>1</sup> It focuses on the computational complexity, because that is the most significant contribution to the power consumption. Based on these numbers and some simulations, the favored correlator-type is chosen. Section 6.2 gives an overview of the different hardware architectures. The hardware architecture is chosen in section 6.3, and elaborated in section 6.4. The mapping of the chosen correlator-type onto the chosen hardware architecture, and simulation results thereof, are described in section 6.5. Finally conclusions are drawn and some recommendations are given.

# 6.1 Digital correlators

There are two main algorithms for digital correlation, and both have their advantages and disadvantages. They are known as the FX-correlator (FXC) and the XF-correlator (XFC). The naming convention stems from the order in which crosscorrelation (denoted by X) and the Fourier transform (denoted by F) are calculated. The FXC calculates the spectrum following eq. (2.15), while the XFC calculates the spectrum following eq. (2.14).

## 6.1.1 FX-correlator

The FXC calculates the cross-spectrum by first transforming the two sequences of samples x[n] and y[n] to the frequency-domain X[f] and Y[f]. This is usually done using the efficient FFT-algorithm. It then takes the complex conjugate  $\overline{Y}[f]$  of Y[f] and calculates the product  $X[f]\overline{Y}[f]$ . The results is a complex cross-spectrum. The process is schematically depicted in fig. 6.1.

Figure 6.1 also shows some extensions to the basic idea, such as windowing the incoming data samples and averaging the results of several spectral measurements, both of which are discussed in chapter 2.

#### 6.1.2 XF-correlator

The XFC calculates the cross-spectrum by first calculating the ccf  $c_{XY}[k]$  and then taking the Fourier transform of  $c_{XY}$ .

<sup>&</sup>lt;sup>1</sup>Note that, in the current system design, each ADC generates *real* numbers, and that each path only has one ADC. For I/Q-mixing, each path would have two ADCs, and the samples from each path need to be considered as complex numbers, which will require some changes in the algorithms discussed.



Figure 6.1: The FXC first calculates the Fourier transforms of the incoming samples to arrive at the cross-spectrum.

Calculating the ccf can be done using e.g. eq. (2.7) or eq. (2.8). Again, extensions to the basic idea, such as windowing the ccf, are discussed in chapter 2.

## 6.1.3 Computational complexity

Deriving computational complexity implies the use of a specific implementation, which is in general not desired when looking at it from a system point of view. However, the basic operations of both correlators are so common that this should not be a problem. The derivation will refrain from delicate optimizations, as this is very implementation-specific. Real arithmetic operations, like additions and multiplications, require a certain amount of dedicated hardware or, if some processing unit is used, a certain number of clock cycles. Complex operations are broken down into real operations, and real operations are broken down into additions (which includes subtractions), multiplications and divisions.<sup>2</sup> Memory requirements are not discussed in this comparison, because neither method requires an excessive amount of memory and the exact amount required is again very implementationspecific. Per functional block of the correlator, the computational complexity will be derived, followed by an overview of the complexity for a complete correlator.

# Phase shifting

As discussed in chapter 5, the two paths may be phase-shifted with respect to each other. A mixer acts as a constant phase-shifter, which can be easily seen by looking at the important first harmonic  $f_{\rm LO}$  of the mixer (the other harmonics of the mixer are undesired anyway and suppressed by the techniques discussed in chapter 4):

$$\left(\sum_{i}\cos(2\pi f_{i}t + \phi_{i})\right) \cdot \cos(2\pi f_{\rm LO}t + \theta) = \frac{1}{2}\sum_{i}\left(\underbrace{\cos\left(2\pi \left(f_{i} - f_{\rm LO}\right)t + \phi_{i} - \theta\right)}_{-\theta \text{ is a constant phase-shift}} + \underbrace{\cos\left(2\pi \left(f_{i} + f_{\rm LO}\right)t + \phi_{i} + \theta\right)}_{\text{undesired, removed}}\right) \quad (6.1)$$

For proper correlation, this constant phase-shift needs to be undone in the digital domain. Since the plan is to use FFTs in the calculation process, a constant phase-shift is just one complex multiplication per output bin of the FFT of the signal that needs to be phaseshifted.

 $<sup>^{2}</sup>$ Some instruction sets, such as SSE3 [96], have instructions suitable for complex arithmetic, but under the hood the operations are still broken down into real operations, so the use of real arithmetic operations provides a better starting point for comparison.

# 6.1. DIGITAL CORRELATORS



Figure 6.2: The XFC first calculates the ccf of the incoming samples to arrive at the cross-spectrum.

## **Complex numbers**

A complex addition is equal to two real additions. The straightforward calculation of complex multiplication requires four real multiplications and two real additions, while a different way of calculating a complex multiplication requires three real multiplications and five real additions [97]:

$$x \cdot y = (a + ib) \cdot (c + id) = (ac - bd) + i(ad + bc) = (ac - bd) + i ((a + b) (c + d) - ac - bd)$$
(6.2)

Which alternative to choose depends on the situation.

#### Windowing

Windowing real samples requires one real multiplication per sample. It is assumed that the window-values themselves do not need to be calculated but are available somewhere in memory.

#### $\mathbf{FFT}$

For an *M*-point complex FFT, where *M* is a power of two and  $2^{L} = M$ , one requires M - 4 real multiplications, 2LM - 4 real additions and  $\frac{1}{2}LM - 2M + 4$  complex multiplications (see section B.11). Note that only the standard radix-2 FFT is considered, which is the most common, but not always the best solution [98]. Because the sequences of samples are real, the fact that the result of the FFT is Hermitian can be exploited. It can then be done in half the number of additions and multiplications [98].

## Complex conjugation

Complex conjugation requires the imaginary part of a complex number to be negated. This could be considered as a subtraction. In that case, complex conjugation requires one subtraction. For the FXC, complex conjugation is only necessary right before a complex multiplication (the XFC does not need conjugation). Similar to regular complex multiplication, these two operations can be rewritten into one:

$$x \cdot \overline{y} = (a+ib) \cdot (c-id)$$
  
=  $(ac+bd) + i(bc-ad)$   
=  $(ac+bd) + i((a+b)(c-d) - ac+bd)$  (6.3)

Complex conjugation in this case does not involve any additional computation.

#### **Crosscorrelation function**

Calculating the ccf is only done in an XF-implementation. To be comparable to the FXcorrelator, 2M - 1 lags need to be calculated. For lag k = 0, one multiplication and one addition per pair of samples can be calculated. For lag  $k = \pm 1$ , the same situation arises, except that the number of pairs of samples is one less than for lag k = 0. In general for lag k, one needs |k| multiplications and additions less than for lag k = 0.

Assume a total of KM samples are available to calculate 2M - 1 lags. Then

$$\sum_{k=-(M-1)}^{M-1} KM - |k| = 2KM^2 - M^2 + M - KM$$
(6.4)

multiplications and additions need to be calculated. Finally, when the correlation is finished, 2M - 1 divisions are required.

## Averaging

Averaging n real numbers requires n-1 real additions and 1 division.

#### 6.1.4 Complexity of the correlators

In the end one desires an M-point spectrum. It is assumed that a total of KM samples per ADC are available.

For the FXC this means that per spectral estimate one needs to window 2M samples, calculate two *M*-point FFTs on real numbers, take the complex conjugate of *M* points and perform *M* multiplications. When averaging *K* spectral estimates, 2(K-1)M additions and

88

Operation	Complex Mult.	Real Mult.	Add.	Div.
Windowing F-part X-part	$\frac{KM(\frac{1}{2}L-2)+4K}{KM}$	$\frac{2KM}{K(M-4)}$	2K(ML-2)	
Averaging			2M(K-1)	2M
Total	$\frac{1}{2}KML + K(4-M)$	3KM - 4K	2KM(L+1) - 4K - 2M	2M

Table 6.1: Computational load of an FXC with KM samples per ADC and an M-point result.

Table 6.2: Computational load of an XFC with KM samples per ADC and an M-point result.

Operation	Complex Mult.	Real Mult.
Windowing F-part X-part Averaging	$M(\frac{1}{4}L - 1) + 2$	$2M - 1 \\ \frac{\frac{1}{2}M - 2}{M^2(2K - 1) + M(1 - K)}$
Total	$M(\frac{1}{4}L - 1) + 2$	$M^2(2K-1) - MK + \frac{7}{2}M - 3$

Table 6.3: Computational load of an XFC with KM samples per ADC and an M-point result.

Operation	Add.	Div.
Windowing		
F-part X-part	LM - 2 $M^2(2K - 1) + M(1 - K)$	2M
Averaging	$\mathcal{M}(\mathcal{Z}\mathcal{M} = 1) + \mathcal{M}(1 = \mathcal{M})$	2111
Total	$M^{2}(2K-1) + M(L-K+1) - 2$	2M

2M divisions are required (the factor 2 comes from the fact that the entities are complex-valued).

For the XFC one only has one spectral estimate, so windowing only needs to be done once. Similarly only one (2M - 1)-point FFT needs to be calculated, which using DIF can be reduced to an *M*-point FFT. It is assumed that the computations involved with the decimation can be neglected. Before calculating the FFT one needs to calculate the (2M - 1)-point correlation function.

Tables 6.1 and 6.3 show the computational load of both correlators, where complex multiplications are not converted to real operations.

Figure 6.3 compares the values from tables 6.1 and 6.3 graphically, where the complex multiplications are rewritten as 4 real multiplications and 2 real additions. Note that a fractional K is possible for the XFC, but not for the FXC (unless overlapping is used, as discussed in chapter 2).

For relatively large M and large K it can be seen that the number of real multiplications (again regarding a complex multiplication as four real multiplications) for the XFC is approx-



Figure 6.3: Comparison of computational load between an XFC and an FXC for two values of M.  $M_{\rm FXC}$  and  $M_{\rm FXC}$  represent the number of real multiplications in these correlators, and  $A_{\rm FXC}$  and  $A_{\rm FXC}$  represent the number of real additions.

imately  $2KM^2$ , while for the FXC it is 2KML. The XFC has to perform  $M/L = M/\log_2 M$  more multiplications than the FXC. For  $M = 2^{10}$  this factor is 100. In fig. 6.3 one also finds this value, even for K as low as 5.

## 6.1.5 Comparison

Although both methods are mathematically equivalent<sup>3</sup>, their implementation and computational loads were shown to differ significantly.

A comparison based on such abstract computational load, however, is not always completely fair. In radioastronomy for example, samples from the telescopes are often only two bits [99]. When an FXC is used, an FFT on 2-bit samples requires 6 bits at the output [99]. The multipliers after the FFT therefore need to have 6-bit inputs, which makes them much bigger and more complex than the two-bit multipliers that can be used in an XFC. For radioastronomy another, far more important, argument often shifts the balance to an XFC-implementation, and that is interconnection costs [99]. Six bits simply require three times more capacity than two bits.

Taking the product of two spectra (as done in the FXC) to generate the correlation causes aliasing in the lag domain. This decreases the signal-to-noise ratio by an estimated factor of 1.22 or equivalently 0.86 dB for spectral line observations [99]. A hybrid form of both correlators, known as an FXF-correlator, also exists, which trades off the advantages of the FXC and the XFC [99]. A more detailed analysis is presented in [99], which also presents some other issues not discussed here. The complicated interaction between all the possibilities in the design of a correlator requires more understanding of all the processes involved and is left for future research.

Interconnection costs are not an issue on-chip, but the number of bits required in each stage is. Finite wordlength effects, as present in practical implementations, will be different for the XFC and the FXC. For now the power consumption of both alternatives is the most interesting part, and it is assumed that the effect on all other properties is negligible.

It was calculated (see chapter 4) that the ADCs need to sample with a resolution of 10 bits. For the XFC this means  $(10 \times 10)$ -bit multipliers need to be used. Rovers [4] already

<sup>&</sup>lt;sup>3</sup>Smoothing the ccf in the XFC cannot be mapped one-to-one to windowing in the FXC, but the effects that one achieves are very similar.

showed that performing an FFT using 16 bits gives the required 70 dB SFDR. In the FXC and XFC, the FFT thus needs to be computed using 16 bits. In the FXC, multiplications need to be performed after the output of the FFT, requiring  $(16 \times 16)$ -bit multipliers.

#### 6.1.6 Digital power consumption

CMOS digital circuitry consumes power, which is commonly divided into static and dynamic power. Static power consists of all kinds of leakage currents, of which subthreshold current and gate leakage current are the main contributors [100, 101]. Dynamic power is subdivided into switching power (currents flowing to charge the load capacitances), short-circuit power (currents flowing from power supply to ground when both the NMOS and PMOS are conducting) and glitches (gates making several transitions before settling), sometimes also referred to as toggle power [102], although some authors consider glitches to be part of the switching power [103] or consider short-circuit power separately [104].

Before the invention of CMOS, mostly only NMOS-transistors were used, which means that half of the time they short-circuited their outputs, with large static power consumption as a result. When CMOS was introduced, static power consumption suddenly became negligible compared to dynamic power consumption. Well into the 1990's, research focused mainly on reducing this dynamic power consumption. Nowadays, with lower voltages and gate lengths and widths decreasing, static power consumption is an important factor again, and it is predicted to dominate within a few years [100, 101]. Optimizations on the peripheral level (coolers to reduce the leakage current), system level (pipeling, multiple voltage supplies), software level (compilers for efficient hardware usage), circuit level (sleep transistors) and technology level (lower supply, multiple threshold voltages, high-k dielectrics) may reduce this power consumption [100].

Static power consumption can be easily determined using simulation when a design is complete, but it is not easy to estimate it beforehand. It can for example be reduced by time-multiplexing parts of the hardware, but depending on other constraints this may not always be possible. Therefore, although static power consumption becomes more and more important, we will focus on dynamic power consumption.

Estimating dynamic power consumption is difficult because it is input-dependent [102]. Several estimation procedures exist [102, 105, 106], many of which rely on the determination of circuit activity (the fraction of clock cycles that gates switch). The switching power consumption can be written as  $P_{\text{switch}} = \alpha f C V^2$ , where f is the clock frequency, C the load capacitance, V the supply voltage and  $\alpha$  the activity [100–104]. Empirical measurements showed that activity on true data ranges from 0.01 to 0.25 [104].

### Dynamic power and minimum feature size

In the expression for  $P_{\text{switch}}$ , the load capacitance depends on the technology used, which is important if circuits, realized in different technologies, are compared. To be able to compare digital circuits produced in different CMOS-technologies, approximate expressions for scaling rules need to be derived.

It is well known that for two parallel plates the capacitance is  $C = \epsilon A/d$ , where  $\epsilon$  is the dielectric constant of the material between the plates, A the area of each plate and d the distance between the plates. To a first-order approximation, the gate-capacitance of a transistor resembles a parallel-plate capacitor.

A reduction of the minimum-feature size by a factor of 2 reduces the capacitance by a factor of 4 because the area A scales by a factor of 4. In older technologies the supply voltage was brought down to reduce power consumption. To prevent short-channel effects from ruining the desired operation of the transistor, the gate-oxide thickness d had to scale down as well. The gate-oxide was scaled down roughly in proportion to the minimum feature-size, which means A/d scaled linearly as well. Starting approximately at 0.13  $\mu$ m CMOS-technology, leakage currents prevented the oxide from being scaled down any further,



Figure 6.4: Raw power consumption results using QuestaSim for the FXC running at 100 MHz.

so from then on A/d scaled quadratically with minimum-feature size (and the supply voltage stayed at 1.2 V).

#### Power consumption of an XF-correlator

An initial design of an XFC was implemented in VHDL. The designed circuit is shown in fig. 6.2, and contains only the correlator (registers and Multiply-Accumulate (MAC)-units). In this implementation the multipliers are  $(8 \times 8)$ -bit (it was constructed and simulated before it became known that 10 bits are required for the multiplications). It is synthesized twice; once with accumulators with a width of 24 bits and 7 MAC-units, and once with accumulators of 48 bits and 257 MAC-units.

The model is synthesized using the Synopsys Design Compiler for a 90 nm TSMC-process operating at 1.2 V. Because power consumption of digital circuits heavily depends on toggle rate (activity) of internal nodes [102, 103, 105, 106], three input streams are created. The first stream contains a slowly varying sine with a period of slightly more than 5215 samples, the second a fast varying sine with a period of slightly more than 31 samples. Non-integer number of samples for a period were chosen to avoid the streams to be repetitive. The third stream contains full-scale uniformly distributed random noise. All streams contain  $2^{14}$  samples.

The power consumption is determined using QuestaSim. The raw results are shown in fig. 6.4. The dynamic power consumption of the fast sine is used for further calculations. The static power consumption clearly is much lower than the dynamic power consumption, which makes our choice to focus on dynamic power consumption justified.

The results need to be scaled to our situation, which is a 65 nm process, 2047 MAC-units (for a 1024-point FFT) and  $(10 \times 10)$ -bit multipliers. All formulas involving scaling use the following order (see table 6.4 for the scaling rules and the meaning of the variables):

$$P_{\text{after scaling}} = P_{\text{before scaling}} \cdot \frac{y_f}{x_f} \cdot \left(\frac{y_v}{x_v}\right)^2 \cdot \frac{y_L}{x_L} \cdot \left(\frac{y_l}{x_l}\right)^2 \cdot \left(\frac{y_m}{x_m}\right)^2 \cdot \frac{y_c}{x_c}$$

where factors are omitted if they are not needed.

The final estimated dynamic power consumption then at 200 MS/s is the average of the

Quantity	Scaling factor
Frequency	$y_f/x_f$
Voltage	$(y_{v}/x_{v})^{2}$
Minimum-length $(x, y \ge 130 \text{ nm})$	$y_L/x_L$
Minimum-length $(x, y \le 130 \text{ nm})$	$(y_l/x_l)^2$
Multiplier-width [107]	$(y_m/x_m)^2$
Correlation-stages	$y_c/x_c$

Table 6.4: Scaling rules for dynamic power consumption when going from quantity x to quantity y.

two powers found:

$$P_{\rm dyn,1} \approx 0.108 \cdot \frac{200}{100} \cdot \left(\frac{65}{90}\right)^2 \cdot \left(\frac{10}{8}\right)^2 \cdot \frac{2047}{257} = 1.40 \ [W]$$
$$P_{\rm dyn,2} \approx 0.0024 \cdot \frac{200}{100} \cdot \left(\frac{65}{90}\right)^2 \cdot \left(\frac{10}{8}\right)^2 \cdot \frac{2047}{7} = 1.14 \ [W]$$

resulting in  $P_{\rm dyn} = P_{\rm dyn,1}/2 + P_{\rm dyn,2}/2 \approx 1.3 \, {\rm W}.$ 

#### Power consumption of an FX-correlator

The main computational load of an FXC consists of calculating the FFTs. Since the samples are real-valued, two real FFTs can be calculated using one complex FFT [98]. At a sample rate of 200 MS/s and 1024-point FFTs, the SA needs to calculate  $2 \times 10^5$  complex FFTs per second.

An Application-Specific Integrated Circuit (ASIC)-implementation of an FFT was made by Heysters [108]. The required energy per 16-bit 1024-point FFT is 1938 nJ using a 120 nm process with a voltage supply of 1.2 V. Following the scaling rules in table 6.4, this boils down to 569 nJ per FFT, or at a sample rate of 200 MS/s, 0.11 W.

Heyster's ASIC-implementation is not optimal in the sense that its architecture resembles the Montium-implementation. A more optimized FFT-processor is discussed in [109], which uses 1240 nJ per 16-bit 1024-point real FFT using a 180 nm 1.2 V process. At 65 nm, 1.2 V this becomes 526 nJ per 1024-point complex FFT, and at 200 MS/s the estimated power consumption is 0.10 W. Note that an 8 times more efficient FFT-processor was designed in [109] using subthreshold operation of CMOS. This is beyond the capabilities of current synthesis tools, and therefore this value is not used.

Using the results from the previous section, one can estimate the required power for the MAC-operations that need to be performed after the FFT. In the XFC,  $M^2(2K-1) + M(1-K)$  MAC-operations are computed (see table 6.3), while in the FXC, KM complex multiplications and additions are computed, or, using four real multiplications per complex multiplication, equivalently 4KM real MAC-operations. The XFC has to perform  $4.1 \times 10^{11}$  MAC-operations per second ( $K = f_s/M$ , with  $f_s = 200$  MHz and M = 1024), while the FXC only needs to calculate  $8.0 \times 10^8$  MAC-operations per second, which is a factor 500 less. The power consumption of these operations for the FXC therefore corresponds to 2.5 mW. Scaling from 10 to 16 bits, this becomes 6.5 mW, indeed insignificant compared to the power required by the FFTs.

With respect to the number of multiplications that need to be performed by the FXC and the XFC, tables 6.1 and 6.3 suggest that for M = 1024 (and thus L = 10) and large K, the power consumption of the FXC is  $2.10 \times 10^6/2.25 \times 10^4 \approx 93$  times less than of the XFC, while the power estimations suggest a factor 1.3 W/0.10 W = 13. Note that, first of



Figure 6.5: Schematic overview of the power-efficiency and flexibility of different hardware architectures.

all, the XFC performs multiplications with 10-bit samples, while the FXC multiplies 16-bit samples. This gives a power consumption reduction of  $(16/10)^2$ , so the net difference would be a factor 36 instead of 93. The dynamic power consumption is very input-dependent as can be seen in fig. 6.4, and the same would hold for the FFT-ASIC. This could easily result in an additional factor 2 difference in the estimation process. Another important aspect may be the wiring; even though the FXC has to perform far fewer multiplications, the wiring is as complex as, if not more complex than, that of the XFC. A factor 13 difference in power consumption therefore seems like a reasonable number.

# 6.2 Hardware architectures

In embedded systems there often is a trade-off in system design between power consumption and flexibility. Different hardware architectures serve different parts of the spectrum.<sup>4</sup> Flexible architectures can be reused for different tasks and are often relatively easy to program, while highly specialized hardware can only do one thing, but can do so with the minimum amount of power required.

An architecture that is flexible and easy to program implies a short time-to-market, but flexibility comes at the cost of higher power consumption, see fig. 6.5. The optimum depends on the application. The four main categories will be very briefly discussed here; a more detailed overview is given in [108]. Note that exceptions define the rule, and that there are many architectures adopting parts from several categories.

# 6.2.1 ASIC

An ASIC is developed specifically for the task at hand, and can be highly optimized. Design, layout and testing costs are very high. Production costs are more or less fixed (in the order of a million euro for a mask set), so it only becomes affordable if produced in large numbers. Since it is purpose-built for the application at hand, it is the most power-efficient solution.

<sup>&</sup>lt;sup>4</sup>Pun intended

#### 6.2. HARDWARE ARCHITECTURES

If any design parameters change, the entire ASIC must be refabricated (which includes designing a new mask set), a process that consumes months and a lot of money.

Many vendors sell ASICs, which, because they are produced in large quantities, are relatively cheap. That is why one finds them in abundance in all kinds of equipment, such as mobile phones, routers and cars. Of course, they still are inflexible.

# 6.2.2 GPP

The General Purpose Processor (GPP) did not get this name without a reason. It can pretty much do anything, which makes it the most flexible solution. The downside is that all this flexibility implies a lot of hardware and control which is not necessary for any single application. It is therefore also the least power-efficient solution.

#### 6.2.3 DSP

Many embedded systems applications involve the processing of large streams of data. The required operations are often quite simple (think of multiplications and additions), but they need to be performed very fast and very often. These processors are optimized for this kind of operations, although a significant part of commercially available Digital Signal Processors (DSPs) can also do a lot of other operations (but not everything a GPP can). These other operations usually require a lot more time than the simple multiplications and additions. Because a DSP is optimized for number crunching, it can do so with a power-efficiency between that of an ASIC and a GPP. The flexibility is also between that of an ASIC and a GPP.

#### 6.2.4 Reconfigurable architectures

Reconfigurable architectures are architectures that can be configured into a mode of operation, but, at the cost of some overhead, can be reconfigured to operate in a different mode. One often distinguishes between fine-grain reconfigurability (bit-level) and coarsegrain reconfigurability (word-level) [108]. An example of a fine-grain reconfigurable device is a Field-Programmable Gate Array (FPGA); an example of a coarse-grain reconfigurable device is the Montium.

# FPGA

An FPGA is a device that, in its elementary form, consists of many Lookup Tables (LUTs), memory elements and interconnects. By describing a digital circuit in a language like VHDL and programming the FPGA with the result, the digital circuit functionality is implemented by connecting the LUTs and memory elements through the interconnects.

Because modern FPGAs have quite a lot of LUTs, many calculations can be performed in parallel. The LUTs are usually only a few bits wide, so many LUTs need to be combined to form the required logic for more complex structures such as  $(16 \times 16)$ -bit multiplications. This is one of the reasons why more expensive FPGAs also contain dedicated arithmetic units and/or DSPs or GPPs.

In short, FPGAs can be reconfigured, can be used for a lot of applications, and are especially good at parallel computations and bit-level manipulation. Once it is configured for a specific application, it is relatively expensive (with respect to time) to reconfigure it for other applications. Furthermore, the relatively long interconnects between all the LUTs and memories make FPGAs power-hungry. Moreover, FPGAs are rather expensive, which makes them a good choice for prototyping, but not for mass-production.

#### Montium

The Montium has a special architecture that can be configured to operate almost as an ASIC, so very efficient but not very flexible. Special care was taken to make the system quickly reconfigurable, so that again it can run almost as efficiently as an ASIC, but then for another algorithm. Its instruction set is comparable to that of a DSP, but only a small subset of the instructions can be used in a single configuration. It combines the efficiency of an ASIC with the flexibility of a DSP.

## 6.3 Design of an FX-correlator

From section 6.1.6 follows that the FXC is much more power efficient. The choice for implementation is therefore easily made.

In section 4.6 the option of also allowing regular spectrum analysis with the proposed SA was discussed. For the implementation, only the crosscorrelation mode is considered here, because that is the main research topic.

# 6.3.1 Signal processing

The digital part is fed with two streams of real-valued samples coming from the ADC. It is assumed that the signal to be measured is available in-phase, which means that phaseshifting is not needed.

The two streams of real-valued samples, denoted by x and y, first need to be translated to the frequency domain by means of an FFT. Since the samples are real-valued, the fact that its Fourier transform is Hermitian can be exploited, i.e. if X represents the Fourier transform of x,  $X[k] = \overline{X[-k]} = \overline{X[M-k]}$ , where the last step follows from the definition of the DFT. Both FFTs can be calculated in one single complex FFT.

Define z[n] = x[n] + iy[n], then [98]

$$X[k] = \text{DFT}(x[n]) = \frac{1}{2} \left( \Re\{Z[k]\} + \Re\{Z[M-k]\} \right) + \frac{i}{2} \left( \Im\{Z[k]\} - \Im\{Z[M-k]\} \right)$$
(6.5)

$$Y[k] = \text{DFT}(y[n]) = \frac{1}{2} \left( \Im \left\{ Z[M-k] \right\} + \Im \left\{ Z[k] \right\} \right) + \frac{i}{2} \left( \Re \left\{ Z[M-k] \right\} - \Re \left\{ Z[k] \right\} \right)$$
(6.6)

where indexing wraps around, i.e. X[M] = X[0]. Note that the conjugate  $\overline{Y[k]}$  can be calculated by negating the imaginary part of Y[k], so it comes at no extra cost.

After the FFTs, the product  $X[k]\overline{Y[k]}$  needs to be calculated. This can be rewritten as

$$X[k]\overline{Y[k]} = \overline{X[M-k]Y[k]} = \overline{X[M-k]}Y[N-k] = \overline{X[M-k]}\overline{Y[M-k]}$$
(6.7)

so this product is also Hermitian, which means X[k]Y[k] only needs to be calculated for  $k = 0 \dots \frac{M}{2}$ .

Define  $a = \Re \{Z[k]\}, b = \Im \{Z[k]\}, c = \Re \{Z[M-k]\}$  and  $d = \Im \{Z[M-k]\}$  for notational convenience. Then

$$X[k]\overline{Y[k]} = \left[\frac{1}{2}(a+c) + \frac{i}{2}(b-d)\right] \cdot \left[\frac{1}{2}(b+d) + \frac{i}{2}(a-c)\right]$$
  
=  $\frac{1}{2}(ad+bc) + \frac{i}{4}(a^2+b^2-c^2-d^2)$  (6.8)

The total process is schematically depicted in fig. 6.6.

Assuming the noise sources and the signal are all uncorrelated, with an input signal S in both paths, a noise source A in path 1 and a noise source B in path 2, one finds

$$X(f)\overline{Y(f)} = (S(f) + A(f))\overline{(S(f) + B(f))}$$
  
= 
$$\underbrace{|S(f)|^2}_{\text{real}} + \underbrace{\overline{S(f)}A(f) + S(f)\overline{B(f)} + A(f)\overline{B(f)}}_{\text{complex}}$$
(6.9)



Figure 6.6: Two real data streams can be correlated using one complex FFT.

In the crosscorrelation one effectively only wants to measure the signal that is present in both paths. Based on the assumption that there is no phase shift between the paths, the wanted result in a real spectrum, so the imaginary part of eqs. (6.8) and (6.9) can be safely disregarded.

Using this route, only  $\frac{1}{2}(ad + bc)$  needs to be calculated, involving 2 multiplications, 1 addition and 1 bitshift. If one would first calculate the complex entities X and  $\overline{Y}$ , and then multiply, it would require 4 additions and 4 bitshifts to obtain X and  $\overline{Y}$ , and then 2 multiplications and 1 addition to calculate  $X\overline{Y}$ . Concluding, the method derived above is more efficient.

# 6.3.2 Choice of hardware architecture

With the discussion of section 6.2, the hardware architecture best fitting to the problem can be determined.

In any SA, user input, for example by manually turning the knobs or by communication using a computer, requires calculations to be performed to determine correction factors, optimum settings, filters to be selected, coefficients to be calculated, etc. Moreover, some of the envisioned applications, such as CR, do not sense the spectrum all of the time. When spectrum analysis is not performed, it would be a waste of chip area and time not to use the computational power available. An architecture that can handle different kinds of tasks is therefore preferable.

On the other hand, power consumption is also very important. First of all, chips require expensive packages or need to be actively cooled in case they produce more than roughly 1 W, which is not preferable because of additional cost and size. Secondly, mobile devices run on batteries, which would quickly drain, if these batteries can deliver the amount of power required in the first place. It was already discussed in section 6.1.6 that a highly optimized ASIC for the FXC would require approximately 0.1 W. It is interesting to see whether a flexible architecture can do the calculations without becoming the dominant power consumer in the entire SA.

Based on [109], a low-power microprocessor requires roughly 40 times more power than the ASIC, which is far beyond reasonable power consumption in the system under design. Hence a GPP is out of the question. An FPGA is not very good at word-level operations, which makes it unsuitable for 16-bit FFT and  $(16 \times 16)$ -bit multipliers as required in our application. The best candidates therefore are DSPs and coarse-grain reconfigurable hardware.

The Montium has 16-bit words and a 32-bit accumulator, and usually performs within a factor of 1 to 5 as compared to ASICs in terms of energy efficiency [110–112], but not always [113]. Based on the assumption that 16-bit FFTs are required, multiplying two FFTs results in 32 bits. The accumulator in the Montium is also 32 bits wide, so in order to allow accumulation, the multiplication of two FFTs needs to be rounded or truncated by several bits, depending on what number of accumulations the design finally allows for. After multiplication of the two FFTs, the values represent power, so in order to have 70 dB of dynamic range at least 24 bits are required.<sup>5</sup> This would allow at least 256 accumulations before the accumulator overflows, which means the Montium may be an option. Unfortunately, in the current version, the output of the accumulator is only 19 bits wide [114], so the Montium cannot be used.

The successor of the Montium, aptly named Montium 2, is currently being designed. In the specifications used in this thesis (October 2, 2008), it can work with 16-bit and 32-bit words, and the accumulator has a width of 40 bits. Its design better resembles a traditional DSP than the Montium 1 does, but it still tries to keep the energy-efficiency of the Montium 1. Therefore it looks like an ideal candidate to map the algorithm to. It is even more interesting because it is still under development, and mapping algorithms to it may help in improving the final design.

#### Alternatives

Choosing the Montium 2 as the first option to map the algorithm to is based on the availability of tooling and available knowledge on this processor. This does not mean that the Montium 2 is necessarily the best option. The range of DSPs and coarse-grain reconfigurable hardware architectures is so large that it would require months to select the most promising candidate.

Some other coarse-grain reconfigurable hardware architectures are discussed in [108]. Of these, the RCP is not designed for low-power, disqualifying it as a good candidate. Promising candidates mentioned there are Pleiades, QuickSilver, XPP and the Avispa. No final conclusions can be drawn from the brief overviews given, and more effort is required to select the right candidates.

There are many different DSP-vendors, of which the most famous are Advanced Devices, ARM, Analog Devices, Freescale Semiconductor, Infineon Technologies, NXP, STMicroelectronics and Texas Instruments. Each of these vendors has a whole range of DSPs, differing in target applications, performance characteristics and power consumption. It is very timeconsuming to find the most appropriate DSP for the current application and has not been attempted.

# 6.4 Montium 2

The Montium 2 is the planned successor of the Montium 1. Both are being developed by Recore Systems, a spin-off of the University of Twente. Whereas the Montium 1 is really a hybrid between an ASIC and a DSP because of its reconfigurability and architecture, the Montium 2 tends much more towards a regular DSP. A block diagram is shown in fig. 6.7.

The architecture of the Montium 2 is not fixed yet; the processor is still under development and only part of the simulation tools work. It is very likely that the memory configuration and the bus structure will be different in the final version. The version used in this thesis dates from October 2, 2008.

The Montium 2 is a 32-bit fixed-point processor with a target clock speed of 200 MHz and a target area of  $1.4 \text{ mm}^2$  in 90 nm technology. Without interconnects, one would expect the area scales roughly quadratically with minimum feature size, giving  $0.7 \text{ mm}^2$  in 65 nm technology. However, including interconnects, it is probably better to assume a safer value of  $1.0 \text{ mm}^2$ . The power consumption is not yet known, and Recore Systems does not yet dare to give an estimate. This issue will be discussed later during the power estimation process.

The processor contains five memories (MEM0 to MEM4), each containing 1024 words of 32 bits, four register banks (RA to RD), each containing 8 words of 32 bits, a loop counter structure for iterative procedures (LC), an 8-bit boolean flag register for software-defined

 $<sup>^{5}70</sup>$  dB corresponds to a factor  $10^{7}$ , which is approximately equal to  $2^{23.25}$ .



Figure 6.7: The Montium 2 architecture (reproduced from [115])

flags (B), an interface to communicate with the outside world (EI and EO), and several different Arithmetic Logic Units (ALUs) (S0 to S3, P0, P1, M0 and M1). Each clockcycle one 32-bit word can be read from some external source, and one 32-bit word can be written to some external sink. It is possible to consider one 32-bit word as two packed 16-bit words.

There are three types of ALUs, which all finish their operations in a single clockcycle. They can all operate in two modes: a mode in which they consider each input as one 32-bit word (the 40-bit words of the S-units are a special case and will be discussed later), and a mode in which each input is considered as two 16-bit words. Their operations are briefly summarized here; for more details see [115].

- Two M-units. Each M-unit has two 32-bit inputs and two 32-bit outputs. Its basic operation is to multiply the lower 16 bits of the two inputs, giving a 32-bit result. When considering the 32-bit input words as two 16-bit numbers, there are effectively four inputs (a, b, c and d). The M-unit can produce the combinations (ab, cd), (ac, bd) or (ad, bc), whatever desired, or simply concatenate without multiplying, giving results as (a|c, b|d) or (a|d, b|c).
- Two P-units. Each P-unit has two 32-bit inputs and one 32-bit output. It can add/subtract the two inputs (with or without saturation and with or without a final single bitshift to prevent overflow) or determine the maximum or minimum. It can also negate the first input, perform bitwise logical functions with the two inputs or determine the exponent.<sup>6</sup> In 16-bit mode, it can combine any 16-bit word from the first input with any 16-bit word from the second input (referred to as a pack-operation,

<sup>&</sup>lt;sup>6</sup>The exponent can be used for dynamic scaling to achieve higher accuracy with fixed-point numbers without the hardware overhead of floating-point numbers, but at the cost of more clock cycles and higher memory requirements.

hence the name P-unit) and perform compare and select operations. Finally, it can also determine the absolute value and do all the arithmetic operations on 16-bit words that it can also do on 32-bit words.

• Four S-units. The S-units are somewhat similar to the P-units. The S-units have a 40-bit accumulator and cannot do pack-operations. They can also perform logical, arithmetic and rotational bitshift on the first input with the number given by the second input.

Note that the 40-bit output of the S-units can only fully be used in any S-unit, because all other wordwidths are 32 bits. None of the S- and P-units provide overflow-bits, because, in Recore's opinion, fixed-point algorithms should be designed without overflow-detection requirements. Possibility of overflow can be detected using the exponent extract instruction, which gives the number of sign extended bits. If this is 0, the next addition could result in overflow. Saturation can be used to mitigate the effects of overflow without overhead.

Code can be written in MontiumC, which can be considered a library with C-functions that map more or less one-to-one onto Montium assembly instructions. Because there is no Montium 2 compiler available yet, MontiumC code cannot be compiled to Montium 2 assembly. MontiumC is plain C, so the code can be compiled to x86-assembly. This provides the possibility to test an algorithm for bugs, functionality and accuracy, but it does not allow scheduling of the ALUs or anything like that, which makes it very hard to estimate the Montium 2 performance when looking at MontiumC code. To assess performance to a higher degree of accuracy, the algorithm should be written in assembly by hand. This can also be compiled to x86-assembly, so it too can be tested. Furthermore, clockcycles have to be explicitly stated in the assembly, so counting clockcycles is possible (although error-prone in case of loops / jumps). At the time of writing, no mechanism is available to automate the counting of clockcycles when executing the algorithm.

# 6.5 Mapping the FX-correlator onto the Montium 2

The design and instruction set of the Montium 2 are not finalized yet, so an implementation in assembly would be quickly outdated. Therefore only an implementation in MontiumC is made. This allows finding bottlenecks in the Montium 2 functionality and the computational accuracy with respect to this particular algorithm. Recore Systems already delivers an efficient radix-2 FFT-implementation both in MontiumC and assembly. Since the main part of the FXC-algorithm consists of calculating FFTs, it is still possible to be pretty accurate with respect to the number of clockcycles required.

In the implementation a few assumptions are made:

- Twiddle factors in the FFT-computation do not have to be calculated but can be read in from some external memory.
- The samples from both ADCs are combined into 32-bit words as shown in fig. 6.8. Because overflow cannot be detected, the FFT-implementation scales the words in every stage. The bits from the ADC are put into the most significant bits of their 16-bit parts to achieve maximum accuracy in the computations. For simplicity it is assumed that the ADCs output their bits in 2-complement form, such that it is directly compatible with the representation used by the Montium 2.
- The 10-bit samples from the ADCs are stored in an external memory. This memory will always have enough free memory available to store the samples required.


Figure 6.8: The samples of the two 10-bit ADCs are combined into a single 32-bit word, which is interpreted as consisting of a 16-bit real part and a 16-bit imaginary part.

Mem 0	Mem 1
0	M/2
1	M-1
2	M-2
M/2 - 1	M/2 + 1

Figure 6.9: The FFT-results are stored out-of-order to allow efficient computation of the correlation.

# 6.5.1 Phase shifting

As discussed on page 86, phase shifting may be required if the two paths in the correlator are phase-shifted. One approach could be to multiply each bin of the FFT of y with the complex factor  $e^{i\theta}$  to acquire a constant phase shift  $\theta$ , see eq. (6.1). Phase shifting has not been implemented in the current implementation.

## 6.5.2 FFT and correlation

A implementation of an FFT, written in MontiumC, accompanies the toolset given by Recore. That implementation writes the final FFT-results in-order to two internal memories: results 0 to  $\frac{M}{2} - 1$  in the first memory and results  $\frac{M}{2}$  to M - 1 in the second memory.<sup>7</sup> In the implementation of the FX-correlator, the final stage outputs the results in a different order (see fig. 6.9). This is done because it allows for easier correlation between the bins.

Calculating the complex multiplications has been combined as in eq. (6.8), where the implementation only calculates the real part and discards the imaginary part. The FFT gives 32-bit outputs, of which both the real part and the imaginary part are 16 bits. It can then be observed that the result of the real part in eq. (6.8) fits in 32 bits because of the multiplication with  $\frac{1}{2}$ . The MontiumC code following eq. (6.8) is shown in listing 6.1, where the line involving the mul\_v-instruction is visualized in fig. 6.10.

The  $\frac{M}{2}$  + 1 results of the correlations are exported to external memory.

 $<sup>^7\</sup>mathrm{Starting}$  counting at zero better corresponds to the nature of the code.

Listing 6.1: MontiumC code for calculating the correlation (binwise multiplication) in the implemented FXC.

```
void binmult(dword M2, dword *mem0, dword *mem1, dword *memres) {
    mul_h(mem0[0],mem1[0],PSL1S,&memres[0],&memres[M2]);
    for (idx i = 1; i < loopidx(M2); i++) {
        mul_v(mem0[i],mem1[i],SWAP_B,&mem0[i],&mem1[i]);
        memres[i]=add(mem0[i],mem1[i],NONE);
    }
}</pre>
```



Figure 6.10: The mul\_v-instruction of listing 6.1 visualized.

# 6.5.3 Averaging

The results of every correlation are stored in external memory, because accumulation after every FFT is not possible. There are four accumulators, which are all used in the FFTprocess. All intermediate accumulation results need to be stored somewhere during the next FFT-computation. All values to accumulate are already 32 bits, and no words of more than 32 bits can be directly stored in local memory or exported to external memory.

After all correlations have been calculated, the averaging process reads in the results one bin at a time and performs the accumulation. Finally it divides the result by the number of correlations and outputs the results to external memory. This is schematically depicted in fig. 6.11.

In the current implementation the number of correlations is given as input. In a general measurement application, it is not known beforehand how many correlations need to be averaged, and a counter needs to be incremented with every FFT. This would also allow averaging any integer number, while in the current implementation only powers of two are allowed. This is done for two practical reasons:

- A division operation requires extra code, and is not easily implemented for the S-units, because a division-step instruction is only available for P-units (a division-step is an instruction needed as part of the divison process). Division by a power of two is a simple bitshift, which can be easily implemented on an S-unit.
- Every factor of two decreases the noise floor by 1.5 dB. It is not likely that any application would require smaller steps in noise reduction.



Figure 6.11: The correlation results are read in from external memory, averaged, and written back.

Listing 6.2: MontiumC code for averaging single correlation measurements in the implemented FXC.

```
void average(dword L, dword logK, dword start) {
   dword K = lsl(int2dword(1),logK); //number of averages
   dword M = lsl(int2dword(1),L); //number of samples per FFT
   dword M2 = lsr(M, int2dword(1)); // M/2
   dword M2_plus1 = add(M2,int2dword(1),NONE); // M/2+1
   dword meme; //external memory address
   dword v; //content of external memory
   dword addr_start = int2dword(0);
   accum acc;
   for (idx j=0; j<loopidx(M2_plus1); j++) {</pre>
      meme=addr_start;
      acc = int2accum(0);
      for (idx i=0; i<loopidx(K); i++) { //accumulate</pre>
         v = rd(meme);
         acc = add_a(acc, v, SAT);
         meme = add(meme,M,NONE);
      }
      acc = asr_a(acc, logK); //divide by #averages (power of 2)
      wr(acc,addr_start);
      addr_start = add(addr_start, int2dword(1), NONE);
   }
}
```

In the case division really is required, it gives a constant overhead, because it only needs to be calculated once for  $\frac{M}{2} + 1$  numbers.

The maximum number of averages is limited by accuracy. Each correlation result contains 32 significant bits. The accumulators have a width of 40 bits, so only 256 32-bit numbers can be accumulated before there is a chance of overflow. It is possible to shift one or more bits to the right to increase the maximum number of averages, but this comes at the cost of accuracy. With the exponent operation, the accuracy can be improved by only shifting when there really is a chance of overflow, i.e. when the Most Significant Bit (MSB) of the accumulator word is unequal to the signal bit, but this comes at the cost of overhead in each cycle to check for the exponent. In the current implementation overflow is not checked; if overflow occurs the results will simply be wrong.

## 6.5.4 Logarithm

The current implementation gives the power spectrum on a linear scale, while it is usually given on a dB-scale. This means the results need to be converted using a logarithm operation. The Montium 2 does not have a logarithm instruction, so it either has to be emulated using the available instructions, or another processor needs to perform this function.

For a stand-alone SA, a user interface needs to be implemented, which generally requires a GPP. It is also likely that a GPP is available in a mobile phone, or that for applications such as BIST the device can be connected to a regular PC. The GPP can then simply handle the logarithm-instructions.

When a GPP is not available, some common techniques to emulate a mathematical operation like the logarithm are the use of a Taylor series approximation and the use of LUTs. In the Taylor series approximation the logarithm is expanded into a polynomial of order n, where n is a design parameter. With higher n the approximation is better but the number of calculations increases.

With a measurement range of (say) 100 dB and a desired accuracy of 0.1 dB, a LUT needs 1000 entries. This fits in the Montium 2, so a LUT-implementation seems feasible. The implementation however is not trivial because of the 32-bit input to the LUT; some (possibly complicated) arithmetic needs to be performed as well to know what entry in the table to select.

An alternative would be to directly map some amount of bits of the input to a memory location, but this results in a very large LUT. A large LUT can be split into several smaller LUTs occupying less total memory [116]. The downside is that the results from each table needs to be added.

It is important to note that only  $\frac{M}{2} + 1$  logarithms need to be calculated, so just like division the time allowed to calculate the logarithm is not really strict. In conclusion, there are several alternatives to a logarithm implementation and more research is required to find a suitable solution.

#### 6.5.5 Memory requirements

Because it is not possible to accumulate on-the-fly, the results of each correlation need to be stored in memory. This memory should be integrated on-chip because it allows a higher bandwidth and lower power consumption [117].<sup>8</sup> With 32-bit results of the multiplication and 40 bits in the accumulator, one can accumulate 256 results without risk of overflow. This means one needs to be able to store  $255 \cdot (M/2 + 1)$  words in external memory. With M = 1024 and each word equal to 32 bits or 4 bytes,  $4.2 \times 10^6$  bits of memory are needed, or equivalently 511 kB.

An estimate of the required chip area for this amount of memory can be made for the two most widely used types of Random Access Memory (RAM): Static RAM (SRAM) and Dynamic RAM (DRAM). The traditional architectures of both types of RAM are shown in fig. 6.12. Many different variations have been proposed, each with its own advantages and disadvantages.

The properties of the SRAM-cells in [118] will be used, as this is a very recent design in 65 nm CMOS operating at 1.2 V. Each cell occupies  $0.54 \ \mu m^2$ , while control overhead is about 4%, so the total area occupied by the memory is 2.36 mm<sup>2</sup>.<sup>9</sup> The leakage current per cell is 750 pA, bringing static power consumption to 3.8 mW. The read current of a cell is 27  $\mu$ A, which, using 32-bit words and constant utilization of the memory, translates to a power consumption of 1.0 mW. Total power consumption of the memory then is 4.8 mW.

 $<sup>^{8}</sup>$ The question is whether this is really necessary, as the Montium 2 only operates at 200 MHz. Furthermore, the samples from the ADCs need to be buffered as well, which is not taken into account here.

<sup>&</sup>lt;sup>9</sup>Each SRAM-cell in the internal memory of the Montium 2 in 90 nm technology currently is about 2.4  $\mu$ m<sup>2</sup>. This would be roughly 1.2  $\mu$ m<sup>2</sup> in 65 nm, so a lot can be gained here with respect to chip area of the Montium 2.



Figure 6.12: The SRAM-cell is about five times as large as a DRAM-cell, but consumes less power.

The area of the memory is quite significant compared to the area occupied by other components of the SA. The SRAM can run at 470 MHz, so it may be possible to decrease the area of each cell, because in the current design only a speed of 200 MHz is required. A better solution could be to use DRAM, as its single transistor plus capacitor per cell occupies an area that is roughly five times smaller than traditional six-transistor-per-cell SRAM [117, 119]. The general thought is that the power consumption of DRAM is higher than that of SRAM, but this is disputed by [117]. Because we do not have any hard numbers on DRAM we will use the SRAM-numbers.

Another option to reduce the memory requirements is to use an additional Montium 2. Even though there will be significant overhead in storing and retrieving the 40-bit accumulator values into and from two 32-bit memory cells, if the Montium 2 only performs this accumulation, it is expected that it can do this real-time. This would completely remove the memory required to store intermediate results. Because the Montium 2 is much smaller than 2.36 mm<sup>2</sup>, it will save chip area, but most likely requires more power.

## 6.5.6 Computational power

The computational power of the Montium 2 with respect to this algorithm needs to be determined by looking at the number of clockcycles required to calculate the different parts of the algorithm.

An FFT-implementation in assembly is given by Recore Systems. It is difficult to see whether the reordering in the last stage requires additional clockcycles, or that it may be calculated at no extra cost. We assume the latter and count the clockcycles of the original implementation to arrive at the cycle count of this slightly changed FFT. These numbers are shown in table 6.5. Reading in twiddle-factors only needs to be performed once (one of the five memories will be constantly reserved for the twiddle-factors), so that is a constant term 3 + M.

The next phase is the correlation. The MontiumC code for the correlation in the current implementation can be mapped one-to-one onto assembly, requiring M/2 clockcycles. Writing the correlation results to external memory costs M/2 + 1 clockcycles, but this can be done simultaneously with the correlation itself. The total number of clockcycles will then be M/2 plus a few additional clockcycles for overhead.

Averaging the results requires external samples to be read in again. With K averages and M points in the FFT this requires KM clockcycles, because only one word can be read in per clockcycle. Accumulation, bitshifting and exporting the result can be done while reading in the next value from external memory. In total averaging requires KM clockcycles plus a few additional clockcycles overhead.

Phase	Clockcycles
Read in twiddle-factors	3 + M
Read in samples	7+M
Initialize FFT	5
First stage FFT	3 + M/2
Middle stages FFT	$(4 + M/2)(\log_2(M) - 2)$
Last stage FFT	1 + M/2
Write samples to ext.mem.	9+M

Table 6.5: Number of clockcycles for different parts of the M-point FFT for the implementation given by Recore.

Table 6.6: Estimated number of clockcycles for different parts of the correlation algorithm.

Phase	Clockcycles	M = 1024	Streaming
Twiddle	3 + M	1027	0
$\mathbf{FFT}$	$(M/2+4)\log_2(M)+8$	6192K	6192K
Crosscorrelate	M/2 + 10	522K	0
Average	KM + 10	1024K + 10	0
	$\mathbf{Stre}$	aming total	6192K

The number of clockcycles can be optimized by simultaneously exporting the results from the correlation to external memory and importing samples from external memory for the next FFT (provided the external memory can handle it). The correlation requires a little over M/2 clockcycles, while reading in the samples requires slightly more than Mclockcycles. The correlation can therefore be calculated without a penalty in clockcycles.

The results are summarized in table 6.6 where an overhead of 10 clockcycles is assumed where no number of clockcycles is known for the overhead. This is expected to be on the safe side.

The streaming total in table 6.6 denotes the number of clockcycles required for doing all calculations except those that only need to be performed once.<sup>10</sup> Operations that only need to be performed once are reading in the twiddle factors before the start of a measurement and averaging the results after the end of a measurement. The streaming total is *not* the sum of the FFT and crosscorrelate entries in the table due to parallelism in the algorithm (see fig. 6.13). Provided enough external memory is available, 32300 1024-point FFTs can be handled per second at a clock frequency of 200 MHz, or equivalently a little over 33 MS/s per ADC.<sup>11</sup>

Assuming K = 256 and the ADCs running at 200 MS/s, only 42 FFTs will be finished when the ADCs have delivered all samples. The others still need to be calculated, which takes 6.62 ms. Finally, the averaging process takes 1.31 ms. So with a maximum of 256 averages, one requires 8 ms after stopping the measurement to arrive at the final results. For lower sampling rates the time will be less.

 $<sup>^{10}</sup>$ For the Montium 1, a streaming implementation of the FFT is available, which greatly reduces the overhead caused by reading and writing samples. At this moment such an implementation is not available for the Montium 2.

<sup>&</sup>lt;sup>11</sup>To handle 200 MS/s real-time, one would require 7 Montium 2's (the algorithm is inherently parallel). In the CRISP-project (http://www.crisp-project.eu) a chip is designed using 9 Montiums, which could do the job.



Figure 6.13: Scheduling of the FXC-algorithm during streaming.

In most applications this additional 8 ms will probably not be a problem. For a standalone SA, pushing a button to stop a measurement will cost more time. When observing Time Division Multiplexing (TDM)-systems, the ADC-samples may not be delivered in a continuous stream, but only in bursts (when a specific sender is sending). The Montium 2 can simply continue the computation during the quiet intervals, reducing the additional computation time even more. The averaging process however will always remain, because that can only be done at the end of a measurement.

Assume a user can wait for 1 s after the measurement, which could be reasonable when using a stand-alone SA for all kinds of measurements. In that case K can be increased to  $2^{15}$ , provided the accumulator can handle it and external memory is large enough.

## 6.5.7 Power consumption

Because the architecture of the Montium 2 is not fully fixed yet, let alone that the hardware has been synthesized, the power consumption has to be estimated based on other architectures.

In section 6.1.6 it was derived that an ASIC, able to handle 200 MS/s real-time, requires 0.10 W.

The Montium 1 requires 577  $\mu$ W/MHz when calculating 1024-point complex FFTs [108]. It was synthesized in 0.13  $\mu$ m CMOS-technology, which makes the power consumption equivalent to 145  $\mu$ W/MHz in 65 nm CMOS. Assuming it can calculate the FFTs in the same number of clock cycles as the Montium 2 can do it, the power consumption for real-time handling (by scaling linear with frequency or by putting multiple Montiums in parallel) is  $145 \times 10^{-6} \cdot 200 \cdot 6192/1024 = 0.17$  W.

A little search on the sites of DSP-manufacturers for DSPs with an architecture comparable to that of the Montium 2 resulted in the TMS320C6421 Fixed-Point Digital Signal Processor from Texas Instruments [120]. It has 64 32-bit registers, six 32/40-bit ALU's, two multipliers, each supporting two (16 × 16)-bit multiplications with 32-bit outputs and 16 kB of internal memory. It is manufactured in 90 nm CMOS and operates at a frequency of 700 MHz with a supply of 1.2 V. Its nominal power consumption is 0.72 W, but it is not known whether this is an accurate figure when it comes to calculating FFTs. It is nevertheless the best estimate currently available. Under the assumption that it can process the algorithm with the same number of clockcycles as the Montium 2, it can handle a stream of  $1024/6192 \cdot 700 \times 10^6 = 115.8$  MS/s. For 200 MS/s the power consumption then becomes  $0.72 \cdot 200/115.8 = 1.24$  W. Scaling down to 65 nm one finally arrives at a power consumption of  $1.24 \cdot (65/90)^2 = 0.65$  W.

According to Recore, the Montium 2 will distinguish itself from other architectures by its energy-efficiency, just as the Montium 1 does. Because the Montium 2 looks more like the TMS320C6421 than the Montium 1, it is estimated that its power consumption will be closer to the power consumption of the DSP from TI than to the power consumption of the Montium 1. We therefore estimate the power consumption through weighted averaging:  $(1 \cdot 0.17 + 2 \cdot 0.65)/3 \approx 0.5$  W.

With only one Montium 2, the power consumption is  $0.5 \cdot 33/200 = 0.08$  W. Compared to the power consumption of the analog components this is not insignificant, but also not

dominating. Implementing the crosscorrelation algorithm using one Montium 2 thus seems like a reasonable solution.

## 6.5.8 Dynamic range

Rovers already showed that 16-bit FFTs manage a dynamic range of about 86 dB. In the current implementation, the multiplication following the FFT is not rounded or truncated, so this should still give a dynamic range of 86 dB. The averaging process is not expected to influence this significantly, because it does not change the range of numbers used. It is therefore expected that the dynamic range of the current implementation of the crosscorrelation algorithm is also around 86 dB.

The MontiumC code is compiled, and the results for different spectra are compared to two reference implementations in Matlab. One reference implementation uses the nonquantized samples, while the other reference implementation uses the same 10-bit quantized samples as those that are fed into the Montium program. Both reference implementations use 64-bit precision floating-point operations for all calculations. The MontiumC results are on a linear scale; converting to dB-scale is done in Matlab's 64-bit floating-point precision. All samples are generated before execution, and are scaled such that the highest absolute value is equal to 1, making full use of all the bits available.

In all simulations three parameters are used: M, the number of points in the FFT, K, the number of averages and NL, the PSD of the uncorrelated noise (relative to that of the sinusoid<sup>12</sup>) injected in each path.

Figure 6.14 shows an example. The signal in each path consists of three components: the sinusoidal signal, correlated band-limited noise and uncorrelated white noise. In this situation, the uncorrelated white noise in both paths has a PSD of -40 dB, just as the band-limited noise that is fully correlated between both paths. With K = 1, one cannot see the band-limited noise due to the high mean value and variance of the white noise. At K = 256, the band-limited noise is clearly visible. The white noise level has gone down by 12 dB, corresponding to the  $\sqrt{K}$  dependence derived in chapter 2.

At some points there is a gap in the black line of the Montium 2. This is because the fixed-point implementation of the algorithm occasionally gives 0 as a result, which in dB becomes  $-\infty$  and cannot be shown in this graph. The difference between the Montium 2 result and the two reference implementations is negligible for this example.

The desired SFDR is 70 dB. In fig. 6.15 an input sinusoid and uncorrelated white noise with a relative PSD of -65 dB is measured. The SFDR is clearly higher than the desired 70 dB.

The maximum SFDR is shown in fig. 6.16, where the initial noise level is chosen low enough. The reference implementation in Matlab using the same quantized samples gives lower values, so this limit is solely due to the fixed-point implementation. The lowest value that can be represented corresponds with -87.19 dB, while the highest value (the peak of the sinusoid) is only 0.001 dB different from the reference floating-point implementation at 0.000 dB. The maximum SFDR therefore is 87.2 dB, very close to the 86 dB observed by Rovers [4].

In reality a sinusoid will never fall exactly into the center of a bin such as in the previous simulations. If its frequency is slightly off, the result is spectral leakage. This leakage is also a limitation to the SFDR. Windows should be used to reduce the leakage, but, as discussed in chapter 2, at the cost of loss in frequency resolution. Some very good windows with a sidelobe suppression of more than 90 dB (such that spectral leakage does not limit SFDR) are discussed in [121].

In chapter 3 the effects of quantization on the SFDR were discussed. Using 8-bit or 9-bit quantization of a full-scale sinusoid gives the result shown in fig. 6.17. In fig. 6.17a the

<sup>&</sup>lt;sup>12</sup>This includes a correction factor for M, because the PSD of a sinusoid scales with the bin-width, while for noise this is not the case.



Figure 6.14: The crosscorrelation algorithm executed on the Montium 2.

highest peak has a magnitude of -65.8 dB, relatively close to the value of -67.9 dB predicted by eq. (3.7) on page 32. For 9-bit quantization an SFDR of 70.5 dB is found, while 75.9 dB is predicted by eq. (3.7). The difference is explained by the fact that now only 1024 points are used in the FFT, such that many harmonics fall into the same bin. In fact, because of the aliasing effect, many harmonics fall directly on top of each other. Unfortunately it is not possible to simulate it with a sinusoid, that does not have a frequency fitting exactly into 1024 samples, because spectral leakage prevents the detection of the small peaks; the current implementation cannot perform a 2<sup>20</sup>-point FFT on the Montium 2 to remove the spectral leakage effect to a negligible level. The same simulation with 10-bit quantization gives a SFDR of 74.3 dB, which justifies the initial choice to use 10-bit quantization.

# 6.6 Conclusions

Of the two general approaches to implement a correlator, the FXC performs much better in terms of computational load and power consumption. Because correlation is not the only digital processing part in a SA, a suitable platform would be a more flexible device than an ASIC. At the same time power consumption is an important issue.

Based on these requirements and the precision needed in the calculations, the Montium 2 is selected as the device to map the algorithm to. The Montium 2 is a 32-bit DSP under development with  $(16 \times 16)$ -bit multipliers running at a target speed of 200 MHz. The implementation of the correlation including FFTs is quite straightforward, but requires external memory because intermediate 40-bit accumulation results cannot be (easily) moved around. 0.5 MB of external memory is required to buffer these intermediate values when 256 averages are made; memory requirements to buffer samples from the ADCs before they can be processed by the Montium 2 are not included. The Montium 2 can process a sample



Figure 6.15: The fixed-point implementation of the crosscorrelation algorithm on the Montium 2 gives an SFDR better than the desired 70 dB.



Figure 6.16: The fixed-point implementation gives a maximum SFDR of 87.2 dB.



Figure 6.17: Quantization shows distortion peaks, which should be well below 70 dB to obtain the desired SFDR.

rate of 33 MS/s per ADC in real-time. Therefore only one Montium 2 is required in the SA. The penalty is an 8 ms of additional processing after the measurement is finished, which is not expected to be a problem in the majority of applications. The estimated chip area of the Montium 2 is  $1.0 \text{ mm}^2$  and its power consumption, based on figures for the Montium 1 and a DSP from Texas Instruments with roughly the same architecture, is estimated to be 0.08 W. The SRAM-memory (unfortunately, no reliable figures for DRAM have been found) requires an estimated chip area of  $2.36 \text{ mm}^2$  with a power consumption of 5 mW.

The correlation algorithm on the Montium 2 limits the SFDR to 87.2 dB as observed in simulations. Other simulations show that 10-bit resolution in the ADC indeed is about the minimum required number of bits to achieve an SFDR of 70 dB when no noise is present.

# 6.7 Recommendations

The current implementation is far from complete, so there are still some parts that need to be worked out. Furthermore, a few changes to the Montium 2 architecture and current tooling are suggested. Finally a general recommendation with respect to measurement time is given.

## 6.7.1 Algorithm

Several parts of the digital processing have been simplified or omitted. This concerns phaseshifting to correct for the relative phaseshift introduced by the mixers, windowing to reduce spectral leakage, division by an arbitrary number instead of a power-of-two (and averaging by a power-of-two) and calculating the logarithm.

In certain situations it is beneficial to average on a logarithmic scale, i.e. one first converts each spectral estimate to dB-scale and then averages the dB-values. For a deterministic signal, linear or logarithmic averaging makes no difference, but for stochastic processes it does. For example, for Gaussian noise the average value using logarithmic averaging is 2.5 dB lower than using linear averaging [122, 123] (at the cost of an increase in variance by a factor 1.64 [123]). Including this option in the software may improve the capability of the SA in measuring small sinusoids, which may be useful in IP3 measurements.

In many SAs the user has to choose parameters such as RBW and the window to use. The processing power available may be used to construct an intelligent analyzer which determines the optimal settings [124].

In the current design the FFTs are windowed and the results are averaged. Windowing gives a fundamental tradeoff between frequency resolution and spectral leakage. An alternative implementation is discussed in [125] which achieves a high dynamic range and at the same time a high resolution, while still being able to efficiently process the data. This alternative looks a lot like the polyphase filter bank implementation discussed in [126]. The downside is a reduced transient response.

Correlation is not always necessary, and with all the analog building blocks available, one can also perform regular spectrum analysis. With two branches, twice as many averages can be obtained in the same measurement time, or one branch can be shutdown to lower power consumption. Performing an FFT on the samples from one branch, one obtains complex numbers which contain phase and amplitude information. An algorithm that can do this conversion is known as the Coordinate Rotation Digital Computer (CORDIC)-algorithm, which is an algorithm that does not use multipliers. Using the availability of multipliers in the Montium 2, an implementation that efficiently combines these with regular CORDIC is discussed in [127]. The accuracy of the phase information also depends on the phase transfer of the analog frontend, so some corrections may be necessary.



Figure 6.18: Setting overflow bits and one additional bit to combine the overflow bits allows for much more efficient checking for overflow.

## 6.7.2 Montium 2 architecture

Because the Montium 2 is still under development, its architecture is not fully fixed yet. Here some suggestions are presented that could make the correlation algorithm perform better. It should be kept in mind that such alterations could make performance worse for other applications, but decisions on that level are up to Recore Systems.

#### Overflow

The current implementation does not check for overflow because that cannot be easily detected. The exponent extract instruction only indicates whether overflow *could* occur, but adding a 32-bit word to the 40-bit signed word  $0100\cdots 00$  will never result in overflow. The net result is that one shifts sooner than strictly required, decreasing numerical accuracy. It may be possible to detect overflow by calculating the addition twice: once with saturation and once without. Comparing the two results can indicate whether an overflow has occurred. Comparing the two results is not trivial, because they are 40-bit wide and all buses are 32-bit wide. The overhead would be substantial.

Our suggestion is to let the ALUs set an overflow bit. The Montium 2 already has a boolean register for flags, so all that needs to be done is add a few more flags which are read-only and set by the ALUs. There are 6 ALUs that can give an overflow, so 6 additional bits in the boolean register are required. An additional optimization may be to add another bit that is an OR of the 6 overflow bits of the 6 ALUs, see fig. 6.18. The optimization lies in the fact that in 99.9% of the time there will be no overflow, and with only one bit to check, the overhead is kept to a minimum. Once overflow is detected, it requires some additional cycles to detect which ALU has overflown.

The same registers can be used for saturation, since any operation can either overflow or saturate, but never both at the same time. The programmer knows which one it is because saturation is explicitly set in an ALU.

### Storing accumulator values

The accumulators internally work with 40 bits, which allows accumulation of 256 32-bit words. There are only four accumulators, while in the correlation algorithm for an M-point FFT M accumulators are required. Of course it is not possible to put 1024 accumulators in the Montium 2, but currently it is not possible to store intermediate results without 8 bit loss of precision. It might be possible (with a significant overhead) to store the 40 bits into two 32-bit memory-cells and put them back into the accumulator with another significant overhead. In the current implementation the decision is made to store the 32-bit values that

need to be accumulated into external memory, such that the entire accumulation can be performed at once.

If one of the five internal memories would be scaled to 40 bits (introducing only a 5% penalty on the total chip-area of the memories) and the required interconnects are scaled to 40 bits (because the interconnect structure is not known, it is not possible to reason about the increase in chip-area or complexity of the bus), it becomes possible to store the words as if they were regular values. With some clever circuitry, the 40-bit memory can also be used as a regular 32-bit memory, such that any algorithm not requiring storage of 40-bit accumulator values is totally oblivient to this change in architecture.

This architectural change may also give the opportunity to occasionally provide a user with an intermediate spectral estimate such that a measurement can be stopped as soon as the user is satisfied.

## External I/O

In the current design the Montium 2 can simultaneously read from and write to the outside world, but both the read and the write operation allow one word to be communicated per clockcycle. In the current implementation many samples need to be read into internal memory for the FFT, many need to be written out into external memory, and finally many values need to be read in again to be accumulated. With a streaming-FFT implementation this overhead may be mitigated, but it is not certain whether such an implementation is possible with the current architecture, so for now the current FFT-implementation is assumed.

With two memories used for the FFT-samples, the speed of reading in samples can be doubled if more words can be read in one clockcycle. The same holds for writing samples to external memory. With one word per clockcycle, only one accumulator can be kept busy, while four accumulators are available. The speed can be quadrupled if four words can be read in per clockcycle.

As an example, consider the situation that two words can be read in at the same time. From tables 6.5 and 6.6, reading in two words at a time reduces the streaming total from 6192K to 5690K, a speed-up of 8%. The averaging process will go down from 1024K to 512K, a speed-up of 50%. In total, the number of clockcycles will go down from roughly 6192K + 1024K = 7216K to 5690K + 512K = 6202K, an improvement of 16%.

Of course, the actual speed-up depends on the external bus and memory architectures. In other words, the network-on-chip needs to be able to handle it. Nevertheless, with the Montium 2 running at a target speed of 200 MHz, it is not unlikely that external buses or memories can run at much higher speeds, allowing a small buffer in front of the memory to provide the Montium with multiple words per clockcycle. Furthermore, it is not unlikely that certain architectures may have several buses and memories. Extending the I/O capabilities of the Montium thus seems like a promising improvement.

#### External memory

The Montium 2 will be designed as intellectual property, i.e. chip designers can include one or more instances of the Montium 2 in their design. If external memory is present to buffer samples as in the current implementation of the correlation algorithm, an option might be to include the memory in the Montium 2 by increasing the number of words in each memory. Each memory now has 1024 words, but as addresses are sent over 32-bit buses, the address space can be much larger. So most likely all that needs to change is the size of the memory and the internals of the A-units (see fig. 6.7).



Figure 6.19: The XFC can be easily accommodated for oversampling by adding delays between taps, in this example for  $\mathcal{O} = 2$ .

# 6.7.3 Montium tooling

Building all the tools required to work efficiently with the Montium is of course a lot of work. With the current tooling, manually written assembly can be compiled and run on a regular PC. Determining the number of clockcycles required to run an application needs to be done by hand and is error-prone. It would be nice if some profiling is possible, even if it only gives the total number of clockcycles for an entire program. This is most likely not too difficult because clockcycles are explicitly given in Montium assembly by writing an empty line. This allows cycle-counting without requiring the full cycle-accurate simulation tooling to be finished.

# 6.7.4 Oversampling

Because the noise floors goes down only slowly, a large measurement time may be required to detect very weak signals. To speed up the measurement process, oversampling may come in handy.

Suppose a signal with a bandwidth B needs to be analyzed. The Shannon-Nyquistcriterium states that  $f_s \geq 2B$  to prevent frequency-aliasing. Under the assumption that the ADCs and the power budget can handle it, it is possible to sample faster than this.

Given an Oversampling Ratio (OSR) of  $\mathcal{O}$ , which means  $f_s = \mathcal{O} \cdot 2B$ ,  $\mathcal{O}$  spectrum averages can be computed in the FXC in the same time by simply using the set of samples  $k \cdot \mathcal{O}$  as one spectral measurement, the set  $k \cdot \mathcal{O} + 1$  as another spectral measurement, etc. until  $k \cdot \mathcal{O} + \mathcal{O} - 1$ . The number of samples for estimating each point in the correlation function as performed in an XFC is also (roughly) multiplied by  $\mathcal{O}$ . The implementation would be rather straightforward in both cases. For (cross)-spectrum averaging one simply requires more memory to store  $\mathcal{O}$  times more samples. For calculating the correlation lags  $\mathcal{O}$  delays need to be present between the taps, as shown in fig. 6.19.

At first sight, this lowers measurement time, or equivalently, allows more noise suppres-

sion in the same amount of time. The drawback is the higher power consumption required by the ADC and the digital signal processing part, but the analog part will consume the same amount of power. As a result the energy consumption for a measurement goes down.

However, several issues can mitigate the benefits of oversampling, or may even make it worse than not oversampling.

First of all, oversampling by a factor of two also doubles the noise bandwidth if no appropriate measures are taken. In that case, the noise floor is increased by 3 dB, which requires four times more samples, while  $\mathcal{O}$  is only 2. So in this case oversampling only decreases performance.

The alternative is to use the same anti-alias filter. In that case the same amount of noise is sampled as in the original case. Unfortunately, because the noise is bandlimited and it is being oversampled, samples are correlated in time. The noise in two branches is still uncorrelated though, so using the asymptotic approximations for the variance of the noise, this should pose no problem. Assuming the correlation in time poses no problem, oversampling seems like a good possibility. Intuitively, correlation in time in each branch (or equivalently a noise acf that does not go to zero fast enough) increases the noise in the ccf estimation because subsequent multiplications tend to have similar values.

Oversampling is not explicitly discussed in [128], but from the text it does look as if oversampling has some effect. More research on this topic is required to understand the effect of oversampling.

# Chapter 7

# Summary & Conclusions

The SFDR of an SA is limited by non-linearity and noise. Improving linearity of the analog part usually introduces additional noise, for example from the additional components used for linearization. In CMOS the SFDR is typically limited to approximately 60 dB for an RBW of 1 MHz and typical mW-range power consumption. For a CMOS-SA, an SFDR of 70 dB is desired to be competitive with commercial SAs.

Crosscorrelation reduces the noise level without affecting linearity, thereby breaking the tradeoff between noise and linearity. It requires the use of two similar measurement paths in which the noise introduced by each path is uncorrelated as much as possible. The uncorrelated noise contributions will average out at the cost of an increase in measurement time. The effect is that the NF of the SA is lowered. The DANL scales with the square root of the measurement time: a two times longer measurement time lowers the DANL by 1.5 dB.

Crosscorrelation in the analog domain is very difficult due to the correlated noise introduced when the two paths are brought together again. For a digital implementation the resolution of the data samples is very important. If the resolution is too low, the SFDR is limited by distortion due to quantization. If the resolution is too high, the ADCs will be too slow and too power-hungry, and the digital hardware needs to be too complex. Novel equations, which are much easier and faster to use than existing analytical formulas, are derived in this thesis, which allow the SFDR to be determined as a function of the resolution of the ADCs. Based on these equations, the ADCs need a resolution of 10 bits. This number was later confirmed in the digital implementation.

A first system design for a frequency range of 0 GHz to 6 GHz has a low-IF architecture, and is optimized for linearity, because crosscorrelation can remove the noise. However, due to the slow reduction of the noise floor through crosscorrelation, the NF of the system is still important. The basic idea is to amplify at IF for two reasons: amplification at RF reduces the effective linearity of subsequent components such as a mixer, and amplification at IF can be made more linear, due to the use of switched capacitors and the higher loopgain that can be achieved using feedback without stability problems.

The SA has a single input, which is split as early as possible to minimize the amount of correlated noise. The two paths after the split are identical. Before the signal enters the integrated SA, it goes through a filterbank. In this external filterbank a number of lowpassor bandpassfilters are present with a (higher) cut-off frequency increasing exponentially per filter. In combination with techniques such as PM and HR-mixing, all undesired harmonics and images should be sufficiently suppressed.

The RF-input is a 50  $\Omega$ -match. In combination with an R–2R-network, variable attenuation with steps of 6 dB is provided. This selectable attenuation is necessary to optimize SFDR, because it is a function of the input power. A Tayloe-mixer, which is very linear, uses this R–2R-network as the resistive part of its *RC*-bandwidth. The IF-circuitry should be made as linear as possible, for example by using switched capacitor techniques. It should

Block	Power [W]	Amount	Total power [W]
Impedance match	< 0.01	1	< 0.01
Attenuator	< 0.01	2	< 0.01
Mixer	0.01	2	0.02
VCO	0.20	1	0.20
IF-circuitry	0.10	2	0.10
ADC	0.04	2	0.08
Correlator	0.08	1	0.08
SRAM Memory	< 0.01	1	< 0.01
Control	< 0.01	1	< 0.01
Total			0.49

Table 7.1: Estimated power consumption of the designed SA per functional block at a sample rate of 200 MS/s.

contain amplification and a suitable interface to the ADC. In the current design a sample rate of 200 MS/s has been chosen, as this is approximately the state-of-the-art sampling speed at the required resolution of 10 bits. The digital crosscorrelation needs to be accurate enough to still allow an SFDR of 70 dB. Of the two possible main architectures for digital crosscorrelation, namely the FXC and the XFC, the FXC is chosen because of its power consumption being more than an order lower than that of the XFC.

For downconversion a VCO is needed, which, for the frequency range of 0 GHz to 6 GHz and a sampling rate of 200 MS/s, needs to be tunable from 100 MHz to 6 GHz. The proposed implementation of the VCO uses two parallel LC-oscillators, one tunable from 8 GHz to 10 GHz, the other from 10 GHz to 12 GHz. Using a few integer frequency dividers, the whole range of desired frequencies can be generated.

The RF-frontend with a frequency range of 0 GHz to 6 GHz has been designed and simulated in 65 nm CMOS. It shows good impedance matching, even when component-spread is taken into account. The simulated IP3 ranges from +21 dBm to +26 dBm as a function of frequency. The simulated NF is 14 dB, which gives an SFDR of approximately 82 dB. With correlation, the NF can be reduced, thereby improving the SFDR. These numbers do not include the noise contributions and linearity limitations of the IF-circuitry, VCO and ADC, so the final SFDR will be somewhat lower. The current figures can compete with those of expensive commercial SAs.

For the digital implementation of crosscorrelation, the Montium 2 architecture was chosen for reasons of power consumption and flexibility. Flexibility is important because crosscorrelation is not the only digital processing required. It was found that the Montium 2 can handle a stream of 33 MS/s per ADC in real-time. At the maximum sampling speed of the ADCs, the additional processing time required, after the actual measurement is finished, is in the order of 8 ms for a 12 dB reduction in DANL, which makes an implementation with only one Montium 2 feasible. Only 0.5 MB of memory is required to buffer intermediate results, but using SRAM the chip area occupied is significant. Unfortunately no reliable values for DRAM were found. The SFDR attainable with the Montium 2 is 87 dB.

Table 7.1 shows the estimated power consumption per functional block at a sampling frequency of 200 MS/s when one VCO is used that obeys the phase noise requirements. The power consumption of the ADC and the correlator scale roughly linear with the sampling frequency. As a rule of thumb, using a standard low-cost package, a chip can dissipate up to about 1 W without thermal problems, which shows that an integrated SA with competitive specifications is possible.

Table 7.2 shows the estimated chip area per functional block. The design is relatively

Block	Area $[mm^2]$	Amount	Total area $[mm^2]$
Impedance match	< 0.01	1	< 0.01
Attenuator	< 0.01	2	< 0.01
Mixer	0.02	2	0.04
VCO	1.33	1	1.33
IF-circuitry	0.08	2	0.16
ADC	0.80	2	1.60
Correlator	1.00	1	1.00
SRAM Memory	2.36	1	2.36
Control	< 0.01	1	< 0.01
Total			6.49

Table 7.2: Estimated chip area of the designed SA per functional block.



Figure 7.1: Tables 7.1 and 7.2 summarized in pie diagrams.

large, but when produced in large quantities, the cost per SA will still be low.

In tables 7.1 and 7.2, techniques such as PM and HR-mixing are not taken into account. These techniques will probably make the power consumption and chip area somewhat higher. Because it will not involve the VCO or ADCs, the increase will most likely not be too significant. Figure 7.1 summarizes the values found.

# 7.1 Conclusions

Crosscorrelation can be used to reduce the noise without affecting linearity, thereby increasing the SFDR of the SA. A two times longer measurement lowers the noise floor by 1.5 dB. Approximations have been derived to estimate the measurement time to observe signals buried in noise.

The correlator is implemented in the digital domain. Because AD-conversion is necessary, the effects of quantization on maximum attainable SFDR were investigated. The SFDR increases by 8 dB/bit for quantization of a sinusoid, and by 6 + 2n dB/bit for quantization of *n* equal-amplitude sinusoids. Gaussian noise added by the analog frontend decorrelates the quantization error from the input signal, decreasing the distortion. If the standard deviation of the noise  $\sigma_N = 1$  LSB, the SFDR increases by 171.5 dB. To be able to obtain an SFDR of 70 dB in every situation, the ADCs should sample with a resolution of 10 bits. The correlator is implemented as an FXC, because it requires less than 10% of the power required by the alternative XFC. At a sample rate of 200 MS/s per ADC, an ASIC can handle the data stream real-time, consuming 0.1 W. The Montium 2 can handle only 33 MS/s real-time, requiring an estimated 0.08 W. Averaging the different spectra can only be done afterwards due to architecture restrictions. This requires 0.5 MB of memory for 256 averages of 1024-point spectra. The additional 8 ms required for processing after the actual measurement is complete is not expected to be a problem, making a solution with one Montium 2 feasible.

A linear RF-frontend, designed for the 0 GHz to 6 GHz region for both a regular SA (with one measurement path) and for a correlation SA, provides impedance matching, variable attenuation to optimize the SFDR, and frequency conversion, but not amplification. Both implementations achieve an IP3 around +23 dBm. The implementation for the regular SA has an NF of 11.2 dB, while the NF of the frontend for the correlation SA is 2.5 dB higher. Both NF and IP3 increase by about 6 dB per attenuation step, although IP3 saturates around +38 dBm due to nonlinearity in NMOS-switches. The achievable SFDR of both implementations is higher than 82 dB, which means, provided the IF-circuitry can be made linear enough, it may not even be necessary to use crosscorrelation at all to obtain an SFDR of 70 dB.

A total system design of the correlation SA was made, giving a total power consumption of 0.5 W (for a sample rate of 200 MS/s) and a total chip area of 6.5 mm<sup>2</sup>. The largest power consumer is the VCO due to the high phase noise requirements, while the largest chip area consumer is the memory required for storing intermediate spectra. These numbers indicate that an all-CMOS implementation of an SA with good specifications seems like a very good possibility.

# 7.2 Future Research

The design of the SA is not complete yet. A number of ideas have been presented to achieve an SFDR of 70 dB with respect to images and harmonics, but they have not been exhaustively covered. Unforeseen problems may be lurking under the surface, so more research is required.

The current design is based on a sample-rate of 200 MS/s, but in certain situations it may be better to sample at a lower sample rate. It is not yet clear what effect this will have on the requirements of the cut-off frequencies of the external filterbank or the RC-bandwidth of the Tayloe mixer.

The external filterbank is probably going to be an expensive part of the SA, and using the frequency-offset technique introduced by Moseley [78] it may be possible that they will not be needed anymore. The images and harmonics will show up as noise, so if they are relatively strong, a much longer measurement time is needed to remove them.

The designed RF-frontend does not include buffers, i.e. the different paths load each other. This makes it non-trivial to include for example I/Q-mixing or techniques such as PM. More research is needed to improve this frontend.

The IF-circuitry has been almost fully neglected in this thesis, even though it is an essential part of the SA. It requires (variable) gain, filtering and an interface to the ADC. There is a great deal of uncertainty about the attainable specifications, especially with respect to linearity and NF. Flicker noise may be a serious concern at low frequencies, not only because of the noise level itself, but also because subsequent samples tend to have similar values [129], which probably implies that correlation will reduce this noise more slowly. Techniques such as switched biasing [130, 131] or subsampling [132] may reduce the effect.

Using a single VCO, the phase noise will be correlated in both branches, which means that its phase noise needs to be very low, requiring a significant percentage of the total power consumption of the SA. An alternative implementation is to use two VCOs locked in

## 7.2. FUTURE RESEARCH

frequency in some way, such that their phase noise is uncorrelated. This may relax phase noise requirements at the cost of measurement time, lowering the power consumption of the VCO.

With respect to linearity, this thesis focuses completely on IP3, because in traditional designs this is usually the limiting factor. However, with the somewhat unconventional architecture presented here, it may very well be that the IP2 becomes the limiting factor. Therefore the attainable IP2 deserves attention as well.

The number of bits in the ADCs was determined purely based on ideal quantization, while practical ADCs suffer from non-ideal effects such as INL and DNL. The practical use of the equations derived in this thesis relating the SFDR and the resolution of a quantizer will be greatly increased if this is taken into account.

The effects of oversampling on the reduction of the noise floor are not clear. It is important to know this, because it may greatly influence measurement time and hence attainable specifications for certain applications. Directly related to oversampling is the real-time handling of the calculations by one or more Montium 2's. There is a definite tradeoff between the number of Montium 2's, the sampling rate of the ADCs, the amount of memory needed to store samples from the ADCs and the intermediate spectral values, the chip area and the total power consumption.

Determining phase relationships between different frequencies of an input signal makes the SA suitable for more applications. Since the hardware and software required for this type of analysis matches that for crosscorrelation to a great extent, it may be possible to incorporate this functionality at a minimum of effort and cost.

# Appendix A

# List of acronyms

- $\textbf{AC} \hspace{0.1in} \text{Autocorrelation}$ method
- $acf \ {\rm autocorrelation}$ function
- AD Analog-to-Digital
- ADC Analog-to-Digital Converter
- **ALU** Arithmetic Logic Unit
- **AR** Auto-Regressive
- ARMA Auto-Regressive Moving Average
- ASIC Application-Specific Integrated Circuit
- BAW Bulk Acoustic Wave
- BIST Built-In Self-Testing
- **BW** Bandwidth
- ccf crosscorrelation function
- CG Conversion Gain  ${\sf CL}\,$  Conversion Loss
- CLT Central Limit Theorem
- **CP** Compression Point
- **CR** Cognitive Radio
- CML Current-Mode Logic
- CMOS Complementary MOS
- CORDIC Coordinate Rotation Digital Computer
- CTFT Continuous-Time Fourier Transform
- DA Digital-to-Analog
- DAC Digital-to-Analog Converter

- DANL Displayed Average Noise Level
- **DIF** Decimation-In-Frequency
- DFT Discrete Fourier Transform
- DNL Differential Non-Linearity
- DRAM Dynamic RAM **DSP** Digital Signal
- Processor **DTFT** Discrete-Time

Fourier Transform

**DUT** Device Under Test

ENOB Effective Number

ESPRIT Estimation of

Signal Parameters via

Rotational Invariance

**EM** Electromagnetic

of Bits

Techniques

Response

FIR Finite Impulse

FFT Fast Fourier

FoM Figure-of-Merit

Field-Programmable

FX Fourier transform -

Multiplication/Correla-

Transform

Gate Array

Processor

**HF** High Frequency

Intermediate

Frequency

**HR** Harmonic Rejection

FPGA

tion

IF

LPF Low-Pass Filter

IL Insertion Loss

Non-Linearity

IP2 Second Order

Input-referred

Intermodulation

Intercept Point

Input-referred

Intermodulation

Intercept Point

IP3 Third Order

**INL** Integral

**LPTV** Linear Periodically Time-Variant

LNA Low-Noise Amplifier

- LSB Least Significant Bit
- MAC Multiply-Accumulate
- MEMS Micro Electro-Mechanical System
- MOS Metal-Oxide-Semiconductor
- MSB Most Significant Bit
- mse mean-squared error
- MUSIC Multiple Signal Classification
- NF Noise Figure
- FXC FX-correlator NMOS n-type MOS GPP General Purpose
  - **OSR** Oversampling Ratio
    - **OFDM** Orthogonal Frequency Division Multiplexing
    - pdf probability density function

- $\ensuremath{\mathsf{PLL}}$  Phase-Locked Loop
- **PM** Polyphase Multipath
- $\mathsf{PMOS}\ \mathrm{p-type}\ \mathrm{MOS}$
- **PSD** Power Spectral Density
- RAM Random Access Memory
- **RBW** Resolution Bandwidth
- **RF** Radio Frequency
- SA Spectrum Analyzer
- SAVG Spectrum Averaging method
- SAW Surface Acoustic Wave
- SH Sample & Hold
- SFDR Spurious-Free Dynamic Range
- SNR Signal-to-Noise Ratio
- SRAM Static RAM
- **TDM** Time Division Multiplexing
- VCO Voltage-Controlled Oscillator
- VSWR Voltage Standing Wave Ratio
- wss wide-sense stationary
- $\textbf{XC} \ \mathrm{Crosscorrelation}$ method
- XF Multiplication/Correlation - Fourier transform
- XFC XF-correlator
- XSA Cross-spectrum Averaging method

LUT Lookup Table MA Moving Average

Appendix B

# Derivations

This appendix contains some derivations of which the results are used in the main text.

# B.1 Expectation of ccf estimator

With a crosscorrelation function estimator

$$c_{XY}[k] = \frac{1}{N} \sum_{n=0}^{N-1} \overline{x[n]} y[n+k]$$

where x[n] and y[n] are considered zero outside the range  $[0 \dots N - 1]$ , the expected value is readily calculated:

$$E[c_{XY}[k]] = E\left[\frac{1}{N}\sum_{n=0}^{N-1}\overline{x[n]}y[n+k]\right]$$
$$= E\left[\frac{1}{N}\sum_{n=\max(0,-k)}^{N-1-\max(0,k)}\overline{x[n]}y[n+k]\right]$$
$$= \frac{1}{N}\sum_{n=\max(0,-k)}^{N-1-\max(0,k)} E\left[\overline{x[n]}y[n+k]\right]$$
$$= \frac{1}{N}\sum_{n=\max(0,-k)}^{N-1-\max(0,k)}\gamma_{XY}[k]$$
$$= \frac{N-|k|}{N}\gamma_{XY}[k]$$
$$= \left(1-\frac{|k|}{N}\right)\gamma_{XY}[k]$$
(B.1)

# **B.2** Covariance of ccf estimator

In Jenkins & Watts [32], pp. 336–337, the covariance of the biased crosscorrelation function (ccf) estimator is calculated for continuous-time correlation, i.e. without sampling. The result is repeated here for convenience:

$$\operatorname{Cov}\left[c_{XY}(\tau_{1}), c_{XY}(\tau_{2})\right] = \frac{T'}{T^{2}} \int_{-T'}^{T'} \phi(r) \left(1 - \frac{|r|}{T'}\right) \,\mathrm{d}r - \frac{T''}{T^{2}} \int_{-T''}^{T''} \phi(r) \left(1 - \frac{|r|}{T''}\right) \,\mathrm{d}r \quad (B.2)$$

where

$$T' \stackrel{\triangle}{=} T - \frac{|\tau_1| + |\tau_2|}{2} \qquad T'' \stackrel{\triangle}{=} \frac{|\tau_2| - |\tau_1|}{2}$$

and

$$\phi(r) \stackrel{\Delta}{=} \gamma_{XX} \left( r - \frac{\tau_2 - \tau_1}{2} \right) \gamma_{YY} \left( r + \frac{\tau_2 - \tau_1}{2} \right) + \gamma_{XY} \left( r + \frac{\tau_1 + \tau_2}{2} \right) \gamma_{YX} \left( r - \frac{\tau_1 + \tau_2}{2} \right) + K(r, \tau_1, \tau_2)$$

where  $K(r, \tau_1, \tau_2)$  is the joint cumulant of the random variables X(t),  $X(t + \tau_1)$ , Y(t + r)and  $Y(t + r + \tau_2)$ .<sup>1</sup> The fourth order cumulant is 0 in case X(t) and Y(t) are Gaussian, which is what was assumed in the main text, and according to [32] can be neglected for non-Gaussian processes.

A direct translation to the time-discrete domain by replacing integrals with summations yields

$$\operatorname{Cov}\left[c_{XY}[k_{1}], c_{XY}[k_{2}]\right] = \frac{N'}{N^{2}} \sum_{n=-N'}^{N'} \phi[n]\left(1 - \frac{|n|}{N'}\right) - \frac{N''}{N^{2}} \sum_{n=-N''}^{N''} \phi[n]\left(1 - \frac{|n|}{N''}\right) \quad (B.3)$$

where

$$N' \stackrel{\triangle}{=} N - \frac{|k_1| + |k_2|}{2} \qquad N'' \stackrel{\triangle}{=} \frac{|k_2| - |k_1|}{2}$$

and

$$\begin{split} \phi[n] &\triangleq \gamma_{XX} \left[ n - \frac{k_2 - k_1}{2} \right] \gamma_{YY} \left[ n + \frac{k_2 - k_1}{2} \right] \\ &+ \gamma_{XY} \left[ n + \frac{k_1 + k_2}{2} \right] \gamma_{YX} \left[ n - \frac{k_1 + k_2}{2} \right] + K[n, k_1, k_2] \end{split}$$

It can be seen that  $\forall k_1, k_2 \in \mathbb{Z}$  the parameters of the correlation-functions in  $\phi[n]$  are integers. At first sight, the summations seem inconsistent, as they iterate through 2N + 1 values for  $k_1 = k_2 = 0$ . However, due to the (1 - |n|/N')-factor, the two extremes are per definition equal to 0. Similar results hold for arbitrary  $k_1$  and  $k_2$ .

Whether replacing the integrals by summations is allowed needs to be verified. Simulations, however, are consistent with these results.

# **B.3** Expectation of cross-spectrum estimators

The general definition of the cross-spectrum estimator is

$$\tilde{C}_{XY}[f] = \text{DFT}\left(\tilde{c}_{XY}[k]\right) = \sum_{n=-(N-1)}^{N-1} \tilde{c}_{XY}[n]e^{-j2\pi fn} = \sum_{n=-(N-1)}^{N-1} w_S[n]c_{XY}[n]e^{-j2\pi fn}$$

where  $\tilde{C}_{XY}(f)$  is the smoothed version of  $C_{XY}(f)$ , with  $w_S[k]$  the smoothing window. In the case where no smoothing is used,  $w_S[k] = 1$  for all  $|k| \leq N$ , with N the number of

126

<sup>&</sup>lt;sup>1</sup>Note that the formula given in [32] contains a sign error for  $\phi(r)$ 

samples taken. Therefore, the expectation is

$$E\left[\tilde{C}_{XY}(f)\right] = E\left[\sum_{n=-(N-1)}^{N-1} w_{S}[n]c_{XY}[n]e^{-j2\pi fn}\right]$$
  
$$= \sum_{n=-(N-1)}^{N-1} w_{S}[n]E\left[c_{XY}[n]\right]e^{-j2\pi fn}$$
  
$$= \sum_{n=-(N-1)}^{N-1} w_{S}[n]\left(1 - \frac{|n|}{N}\right)\gamma_{XY}[n]e^{-j2\pi fn}$$
  
$$= \sum_{n=-(N-1)}^{N-1} w_{T}[n]\gamma_{XY}[n]e^{-j2\pi fn}$$
  
$$= DFT\left(w_{T}[n]\gamma_{XY}[n]\right)$$
  
$$= W_{T}(f) * \Gamma_{XY}(f)$$
  
(B.4)

where  $w_T[k]$  can be regarded as the total window, which includes both the bias and the smoothing window. In case there is no smoothing,  $W_T(f) = W_B(f)$ , with  $W_B(g)$  the Discrete Fourier Transform (DFT) of the triangular bias (Bartlett-window), and eq. (2.16) is found.

# B.4 Variance of cross-spectrum estimator

The covariance of the spectral estimator  $C_{XY}(f)$  and its smoothed version  $\tilde{C}_{XY}(f)$  for continuous-time are given by Jenkins & Watts [32], pp. 414–418. In the calculations for the smoothed version they make an approximation that is not valid everywhere in our situation. Therefore a few of the steps will be repeated here.

The final exact formulation given by [32] for the covariance of a cross-spectrum estimator is

$$Cov \left[C_{IJ}(f_{1}), C_{KL}(f_{2})\right] = \int_{-T}^{T} \int_{-T}^{T} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \frac{2}{T^{2}} \frac{\sin 2\pi f(T - |\tau_{1}|)}{2\pi f} \frac{\sin 2\pi f(T - |\tau_{2}|)}{2\pi f} \times \left(\Gamma_{IK}(f + g)\Gamma_{JL}(f - g)e^{j2\pi g(\tau_{1} - \tau_{2})} + \Gamma_{IL}(f + g)\Gamma_{JK}(f - g)e^{j2\pi g(\tau_{1} + \tau_{2})}\right) \times e^{-j2\pi (f_{1}\tau_{1} + f_{2}\tau_{2})} df dg d\tau_{1} d\tau_{2}$$
(B.5)

which can be rewritten as

$$\begin{aligned} & \text{Cov} \left[ C_{IJ}(f_1), C_{KL}(f_2) \right] = \\ & \frac{1}{T^2} \int_{-\infty}^{\infty} \Gamma_{IK}(x) \frac{\sin \pi T(f_1 - x)}{\pi(f_1 - x)} \frac{\sin \pi T(f_2 + x)}{\pi(f_2 + x)} \, \mathrm{d}x \\ & \times \int_{-\infty}^{\infty} \Gamma_{JL}(-y) \frac{\sin \pi T(f_1 + y)}{\pi(f_1 + y)} \frac{\sin \pi T(f_2 - y)}{\pi(f_2 - y)} \, \mathrm{d}y \\ & + \frac{1}{T^2} \int_{-\infty}^{\infty} \Gamma_{IL}(x) \frac{\sin \pi T(f_1 - x)}{\pi(f_1 - x)} \frac{\sin \pi T(f_2 - x)}{\pi(f_2 - x)} \, \mathrm{d}x \\ & \times \int_{-\infty}^{\infty} \Gamma_{JK}(-y) \frac{\sin \pi T(f_1 + y)}{\pi(f_1 + y)} \frac{\sin \pi T(f_2 + y)}{\pi(f_2 + y)} \, \mathrm{d}y \end{aligned}$$
(B.6)

Now, the first simplification is made by assuming that the spectra are approximately constant over the range  $f_1$  to  $f_2$ , allowing the  $\Gamma$ -factors to be taken outside of the integral.

Since we are only concerned with the variance, i.e.  $f_1 = f_2$ , this is not a problem. The result then is given for continuous-time:

$$\operatorname{Cov}\left[C_{IJ}(f_{1}), C_{KL}(f_{2})\right] \approx \Gamma_{IK}(f_{1})\Gamma_{JL}(-f_{1}) \left(\frac{\sin \pi T(f_{1}+f_{2})}{\pi T(f_{1}+f_{2})}\right)^{2} + \Gamma_{IL}(f_{1})\Gamma_{JK}(-f_{1}) \left(\frac{\sin \pi T(f_{1}-f_{2})}{\pi T(f_{1}-f_{2})}\right)^{2} \quad (B.7)$$

and discrete time:

$$\operatorname{Cov}\left[C_{IJ}[f_{1}], C_{KL}[f_{2}]\right] \approx \Gamma_{IK}(f_{1})\Gamma_{JL}(-f_{1}) \left(\frac{\sin \pi N(f_{1}+f_{2})}{N\sin \pi(f_{1}+f_{2})}\right)^{2} + \Gamma_{IL}(f_{1})\Gamma_{JK}(-f_{1}) \left(\frac{\sin \pi N(f_{1}-f_{2})}{N\sin \pi(f_{1}-f_{2})}\right)^{2}$$
(B.8)

With I = K = X, J = L = Y and  $f_1 = f_2$ , this simplifies to

$$\operatorname{var}\left(C_{XY}[f]\right) = \Gamma_{XX}(f)\Gamma_{YY}(f)\left(\frac{\sin 2\pi Nf}{N\sin 2\pi f}\right)^2 + \left|\Gamma_{XY}(f)\right|^2 \tag{B.9}$$

which is the result given in eq. (2.17), and also found in Proakis [28] for the autocorrelation case.

For the smoothed spectral estimator the last exact formula given by [32] is

$$\operatorname{Cov}\left[\tilde{C}_{IJ}(f_{1}), \tilde{C}_{KL}(f_{2})\right] = \int_{-T}^{T} \int_{-T}^{T} w(\tau_{1})w(\tau_{2})\operatorname{Cov}\left[\tilde{c}_{IJ}(\tau_{1}), \tilde{c}_{KL}(\tau_{2})\right] \\ \times e^{-j2\pi(f_{1}\tau_{1}+f_{2}\tau_{2})} \,\mathrm{d}\tau_{1} \,\mathrm{d}\tau_{2} \quad (B.10)$$

where  $w(\tau)$  is the smoothing window used. If  $w(\tau) = 1$  for all  $\tau$  the results for the nonsmoothed version follow.

At this point, Jenkins & Watts [32] make a simplification based on the fact that  $w(\tau) = 0$ for  $|\tau| > M$ , with  $M \ll T$ . The reason is that in that case the triangular bias  $1 - |\tau|/T$ is approximately a constant 1 for  $|\tau| \ll T$ . This means that in the frequency domain the bias  $(\sin(2\pi fT)/(\pi fT))$  as used in eq. (B.3) can be approximated by  $\delta(f)/2T$ , and therefore eq. (B.10) simplifies to

$$\operatorname{Cov} \left[ c_{IJ}(\tau_1), c_{KL}(\tau_2) \right] = \frac{1}{T} \int_{-\infty}^{\infty} \Gamma_{IK}(g) \Gamma_{JL}(-g) e^{j2\pi g(\tau_1 - \tau_2)} + \Gamma_{IL}(g) \Gamma_{JK}(-g) e^{j2\pi g(\tau_1 + \tau_2)} \, \mathrm{d}g \quad (B.11)$$

This simplification then results in

$$Cov\left[\tilde{C}_{IJ}(f_1), \tilde{C}_{KL}(f_2)\right] \approx \int_{-T}^{T} \int_{-T}^{\infty} \frac{w(\tau_1)w(\tau_2)}{T} e^{-j2\pi(f_1\tau_1+f_2\tau_2)} \\ \times \left(\Gamma_{IK}(g)\Gamma_{JL}(-g)e^{j2\pi g(\tau_1-\tau_2)} + \Gamma_{IL}(g)\Gamma_{JK}(-g)e^{j2\pi g(\tau_1+\tau_2)}\right) \,\mathrm{d}g \,\mathrm{d}\tau_1 \,\mathrm{d}\tau_2 \qquad (B.12)$$
$$= \frac{1}{T} \int_{-\infty}^{\infty} W(f_1 - g)\{\Gamma_{IK}(g)\Gamma_{JL}(-g)W(f_2 + g) \\ + \Gamma_{IL}(g)\Gamma_{JK}(-g)W(f_2 - g)\} \,\mathrm{d}g$$

where W(f) is the Fourier transform of  $w(\tau)$ . Although this may be a reasonable approximation in case many samples are available, it definitely does not suffice in the general case, i.e. if one is interested in any relation between M and N (of course M is never larger than N).

The next simplification made by [32] is the assumption that the  $\Gamma$ -factors are smooth over the width of the spectral window, because in that case they can be taken outside of the integrals. The result then is

$$\operatorname{Cov}\left[\tilde{C}_{IJ}(f_{1}), \tilde{C}_{KL}(f_{2})\right] \approx \frac{\Gamma_{IK}(f_{1})\Gamma_{JL}(-f_{1})}{T} \int_{-\infty}^{\infty} W(f_{1}-g)W(f_{2}+g) \,\mathrm{d}g + \frac{\Gamma_{IL}(f_{1})\Gamma_{JK}(-f_{1})}{T} \int_{-\infty}^{\infty} W(f_{1}-g)W(f_{2}-g) \,\mathrm{d}g \quad (B.13)$$

The assumption that the spectra are smooth is somewhat of a restriction in spectrum analysis, because stable oscillators and sharply filtered signals have quite abrupt changes in their spectrum. The results given by these formulas are not perfectly valid at or near such an abrupt change. A practical solution is to increase the frequency resolution of the window, for example by using more points in the Fast Fourier Transform (FFT).

Setting  $f_1 = f_2$ , the factor  $\int_{-\infty}^{\infty} W(f_1 - g)W(f_2 - g) \, dg$  can be rewritten:

$$\int_{-\infty}^{\infty} W^2(f-g) \,\mathrm{d}g = \int_{-\infty}^{\infty} W^2(g) \,\mathrm{d}g = \int_{-\infty}^{\infty} w^2(\tau) \,\mathrm{d}\tau = \int_{-M}^{M} w^2(\tau) \,\mathrm{d}\tau \tag{B.14}$$

where the transition from W to w is by virtue of Parseval's theorem. Also setting I = K = Xand J = L = Y, the variance of the smoothed spectral estimator becomes

$$\operatorname{var}\left(\tilde{C}_{XY}(f)\right) = \frac{1}{T} \left( \Gamma_{XX}(f) \Gamma_{YY}(f) \int_{-\infty}^{\infty} W(f-g) W(f+g) \,\mathrm{d}g \right) \\ + \frac{1}{T} \left( \left| \Gamma_{XY}(f) \right|^2 \int_{-M}^{M} w(\tau) \,\mathrm{d}\tau \right) \quad (B.15)$$

If the window is narrow enough the integral  $\int_{-\infty}^{\infty} W(f-g)W(f+g) dg$  tends to zero (except at f = 0). When converted to the discrete-time situation and setting X = Y one finds eq. (2.24).

# B.5 Asymptotic properties of SAVG

The expectation of the cross-spectrum was given in eq. (2.16) and repeated here for convenience:

$$E\left[C_{XY}(f)\right] = W_B(f) * \Gamma_{XY}(f)$$

For a relatively large number of samples per spectral estimate, the Bartlett-window tends to a delta-function, and hence eq. (2.16) reduces to

$$E[C_{XY}[f]] \approx \Gamma_{XY}(f)$$

The variance as given in eq. (2.17) (repeated here for convenience)

$$\operatorname{var}\left(C_{XY}(f)\right) = \Gamma_{XX}(f)\Gamma_{YY}(f)\left(\frac{\sin 2\pi fN}{N\sin 2\pi f}\right)^2 + \left|\Gamma_{XY}(f)\right|^2$$

for large number of samples reduces to

$$\operatorname{var}\left(C_{XY}[f]\right) \approx \left|\Gamma_{XY}(f)\right|^2$$

When the total number of samples taken is N, K = N/M averages of the estimate can be made. Averaging K times simply reduces the variance by a factor K and leaves the asymptotic expectation intact. The asymptotic variance of the Spectrum Averaging method (SAVG) is

$$\operatorname{var}(C_{XY}[f]) \approx \frac{1}{K} |\Gamma_{XY}(f)|^2$$

Note that these results suggest that for frequencies where there is no correlation between X and Y the variance will be zero. That is unacceptable when the Cross-spectrum Averaging method (XSA) is used to lower the noise, because it is also very important to know the noise level as a function of the number of averages at those frequencies. This issue will be discussed in section B.6, but here the equations are stated only for SAVG (i.e. with Y = X).

$$E\left[C_{XX}[f]\right] \approx \Gamma_{XX}(f) \tag{B.16}$$

$$\operatorname{var}\left(C_{XX}[f]\right) \approx \frac{1}{K} \left|\Gamma_{XX}(f)\right|^2 \tag{B.17}$$

# B.6 Asymptotic properties of XSA

Obtaining asymptotic results for XSA turns out to be rather tricky. Two sources have been found discussing the problem at hand, at first sight giving quite different results for our system. We will first show the results of both methods, and then link them together.

## B.6.1 The approach of Jenkins & Watts

The asymptotic expectation of spectral estimation using crosscorrelation is given by Jenkins & Watts [32] in a rather complicated way. This is necessary because crosscorrelation of two arbitrary signals yields a relation between the amplitudes and the phases.

The cross-spectrum  $\Gamma_{XY}$  can be split into a *co-spectrum*  $\Lambda_{XY}$  and a *quadrature spectrum*  $\Psi_{XY}$ :

$$\Gamma_{XY}(f) = \Lambda_{XY}(f) - j\Psi_{XY}(f)$$

or in 'polar' form into a cross amplitude spectrum  $\alpha_{XY}$  and a phase spectrum  $\phi_{XY}$ 

$$\alpha_{XY}(f) = |\Gamma_{XY}(f)| = \sqrt{\Lambda_{XY}^2(f) + \Psi_{XY}^2(f)}$$
(B.18)  
$$\phi_{XY}(f) = \arctan\left(-\frac{\Psi_{XY}(f)}{\Lambda_{XY}(f)}\right)$$

A convenient factor is the squared coherency, defined as

$$\kappa_{XY}^2(f) = \frac{\alpha_{XY}^2(f)}{\Gamma_{XX}(f)\Gamma_{YY}(f)}$$

The estimator of  $\Gamma_{XY}$  was already introduced as  $C_{XY}$  in eq. (2.14).  $A_{XY}$  will be defined as the estimator for estimating  $\alpha_{XY}$ . Smoothing the result yields another estimator  $\tilde{C}_{XY}$ . It can be shown that, under the assumption that the total number of samples is so much larger than the number of lags used for the spectral estimation, the influence of the Bartlettwindow becomes negligible [32, p. 375, eq. (9.2.4)]<sup>2</sup>

$$E\left[\tilde{C}_{XY}\right] \approx \int_{-\frac{1}{2}}^{\frac{1}{2}} W_S(f) \Gamma_{XY}(f-g) \,\mathrm{d}g$$

which will be defined as  $\tilde{\Gamma}_{XY}$ .

 $<sup>^{2}</sup>$ Note that under this assumption the same asymptotic approximations for all three ccf-estimators discussed in chapter 2 are found.

Similarly, the smoothed version of  $A_{XY}$ ,  $\tilde{A}_{XY}$ , is introduced as the estimator for  $\tilde{\alpha}_{XY}(f) = |\tilde{\Gamma}_{XY}(f)|$ . It can be shown that [32]

$$E\left[\tilde{A}_{XY}\right] \approx \alpha_{XY}$$
 (B.19)

Jenkins & Watts further show that

$$\operatorname{var}\left(\tilde{A}_{XY}\right) \approx \frac{I}{2N} \alpha_{XY}^2 \left(1 + \frac{1}{\kappa_{XY}^2}\right) \tag{B.20}$$

where

$$I = \sum_{m} w_{S}^{2}[m] = \int_{-\frac{1}{2}}^{\frac{1}{2}} W_{S}^{2}(g) \,\mathrm{d}g$$

Using the system model of chapter 2 and eqs. (B.18) and (B.19) these equations can be rewritten as

$$E\left[\tilde{A}_{XY}\right] \approx \left|\tilde{\Gamma}_{XY}\right| \approx \Gamma_{SS}$$

and with eq. (B.20)

$$\operatorname{var}\left(\tilde{A}_{XY}\right) \approx \frac{I}{2N} \left( \left| \Gamma_{XY} \right|^2 + \Gamma_{XX} \Gamma_{YY} \right) = \frac{I}{2N} \left( 2\Gamma_{SS}^2 + \Gamma_{SS} \Gamma_{AA} + \Gamma_{SS} \Gamma_{BB} + \Gamma_{AA} \Gamma_{BB} \right)$$

# B.6.2 The approach of Briaire & Vandamme

Briaire & Vandamme [133] derived approximations in two extreme cases, one where the signal dominates the noise ( $\Gamma_{SS} \gg \Gamma_{AA}, \Gamma_{BB}$ ), and one where the noise dominates the signal ( $\Gamma_{SS} \ll \Gamma_{AA}, \Gamma_{BB}$ ). During a crosscorrelation measurement the signal will start to dominate at some point, so neither of the extremes provides an accurate approximation. Using interpolation, this error between the real situation and the two extreme cases is mitigated. The result is (where the estimator of the cross-spectrum is denoted by  $S_{XY}$  following [133]):

$$E\left[S_{XY}^2\right] = \frac{K+1}{K}\Gamma_{SS}^2 + \frac{1}{K}\left(\Gamma_{SS}\Gamma_{AA} + \Gamma_{SS}\Gamma_{BB} + \Gamma_{AA}\Gamma_{BB}\right) \tag{B.21}$$

and

$$E\left[S_{XY}\right] \approx \sqrt{\Gamma_{SS}^2 + \frac{\beta}{K} \left(\Gamma_{SS}\Gamma_{AA} + \Gamma_{SS}\Gamma_{BB} + \Gamma_{AA}\Gamma_{BB}\right)} \tag{B.22}$$

with  $\beta$  providing the interpolation between the two extremes:

$$\beta = \frac{\pi}{4K} \left( \frac{\Xi \left( K + \frac{1}{2} \right)}{\Xi(K)} \right)^2 \left( 1 - \frac{\Gamma_{SS}^2}{E \left[ S_{XY}^2 \right]} \right) + \frac{1}{2} \frac{\Gamma_{SS}^2}{E \left[ S_{XY}^2 \right]}$$
(B.23)

where

$$\Xi(x) = \int_0^\infty e^{-t} t^{x-1} \,\mathrm{d}t$$

is the mathematical Gamma-function, but written as  $\Xi$  to avoid confusion with the spectra. The variance can then be calculated using the well-known formula

$$\operatorname{var}(S_{XY}^2) = E[S_{XY}^2] - E^2[S_{XY}]$$
 (B.24)

 $\beta$  will always be between  $\frac{1}{2}$  (for the extreme  $\Gamma_{SS} \gg \Gamma_{AA}, \Gamma_{BB}$ ) and  $\frac{\pi}{4}$  (for the extreme  $\Gamma_{SS} \ll \Gamma_{AA}, \Gamma_{BB}$ ), so for back-of-the-envelope calculations one can use e.g.  $\beta = \frac{2}{3}$  or  $\sqrt{\beta} = \frac{4}{5}$ , whichever comes in handy.

## B.6.3 Linking the two approaches

In the specific case that there is only noise and no signal in the crosscorrelation, one can use the results from Jenkins & Watts [32] to arrive at the results of Briaire & Vandamme [133]. Starting with the latter, eq. (B.22) simplifies to

 $E[S_{XY}] \approx \sqrt{\beta} \sqrt{\frac{1}{K}} \sqrt{\Gamma_{AA} \Gamma_{BB}}$  (B.25)

while eq. (B.24) simplifies to

$$\operatorname{var}(S_{XY}) \approx (1-\beta) \frac{1}{K} \Gamma_{AA} \Gamma_{BB}$$
 (B.26)

Proceeding with Jenkins & Watts, at frequencies where there is only noise,  $E\left[\tilde{A}_{XY}\right] = 0$ , but because there is a variance, negative and complex values are also possible. For a large number of samples, the Central Limit Theorem (CLT) states that the distribution of  $\tilde{A}_{XY}$ is approximately Gaussian. Taking the absolute value of the result yields a  $\chi$ -distribution with one degree of freedom. Although this allows immediate calculation of the mean and variance [134], it can also be directly derived:

$$E\left[\left|\tilde{A}_{XY}\right|\right] \approx \int_{-\infty}^{\infty} |x| \frac{1}{\sqrt{2\pi\sigma_{\tilde{A}}^2}} e^{-\frac{(x-\mu_{\tilde{A}})^2}{2\sigma_{\tilde{A}}^2}} dx$$
$$= \sqrt{\frac{2}{\pi}} \sqrt{\frac{I}{2N}} \sqrt{\Gamma_{AA}\Gamma_{BB}} \approx \sqrt{\frac{2}{\pi}} \sqrt{\frac{1}{K}} \sqrt{\Gamma_{AA}\Gamma_{BB}} \quad (B.27)$$

where  $\mu_{\tilde{A}}$  denotes  $E\left[\tilde{A}_{XY}\right]$  and  $\sigma_{\tilde{A}}^2$  denotes var  $\left(\tilde{A}_{XY}\right)$ . In the last step, the factor I/2N is simplified to 1/K using  $I \approx 2M$  and M/N = K. Calculation of the variance can be done in a similar way:

$$E\left[\left|\tilde{A}_{XY}\right|^{2}\right] \approx \int_{-\infty}^{\infty} \left|x\right|^{2} \frac{1}{\sqrt{2\pi\sigma_{A}^{2}}} e^{-\frac{\left(x-\mu_{A}\right)^{2}}{2\sigma_{A}^{2}}} \,\mathrm{d}x = \frac{I}{2N}\Gamma_{AA}\Gamma_{BB}$$

resulting in

$$\operatorname{var}\left(\left|\tilde{A}_{XY}\right|\right) = E\left[\left|\tilde{A}_{XY}\right|^{2}\right] - E^{2}\left[\left|\tilde{A}_{XY}\right|\right]$$
$$\approx \left(1 - \frac{2}{\pi}\right)\frac{I}{2N}\Gamma_{AA}\Gamma_{BB} \approx \left(1 - \frac{2}{\pi}\right)\frac{1}{K}\Gamma_{AA}\Gamma_{BB} \quad (B.28)$$

Substituting  $S_{XY}(f)$  by  $|\tilde{A}_{XY}|$  in eqs. (B.21)–(B.24) gives very similar results for the formulas of Briaire & Vandamme [133] and the adapted formulas of Jenkins & Watts [32]. The only differences are the constant factors. Note that Briaire & Vandamme use XSA for determining the cross-spectrum, while Jenkins & Watts use the Crosscorrelation method (XC). It is shown in chapter 2 that without averaging or smoothing these two methods are equivalent.

If K = 1 and  $\Gamma_{SS} = 0$ , one finds from eq. (B.23) that  $\beta = \pi^2/16$ . For the expectation, eq. (B.27) has a factor  $\sqrt{2/\pi} \approx 0.798$ , which is very close to the factor  $\sqrt{\beta} = \pi/4 \approx 0.785$  in eq. (B.25). For the variance, eq. (B.28) has a factor  $(1 - \frac{2}{\pi}) \approx 0.363$ , which is again very close to the factor  $(1 - \beta)$  for K = 1 in eq. (B.26):  $(1 - \pi^2/16) \approx 0.383$ .

Because the results of [32] and [133] match so well for this simple situation, and the system model of [133] resembles the system model of chapter 2 perfectly well, we will adopt the approximations of Briaire & Vandamme [133].

Table B.1: Input impedance calculated for connecting an infinite impedance to a branch.

$Z_{\rm in}[n]$ using branch					
1	2	3	4	5	6
$\frac{2}{1}R$	$\frac{6}{5}R$	$\frac{22}{21}R$	$\frac{86}{85}R$	$\frac{342}{341}R$	$\frac{1366}{1365}R$

### B.7 Oscillator power

A general model for oscillator phase noise is discussed in [135]. The most important quantity is the single sideband noise spectral density  $\mathcal{L}\{\Delta\omega\}$  in dBc/Hz, which is a normalized quantity based on the shape of the phase noise [4, 135].

The oscillator number is defined as

$$N_{\rm osc} = 10 \log \left( \mathcal{L} \left( \Delta f \right) \left( \frac{\Delta f}{f_{\rm osc}} \right)^2 \right) \, \left[ dBc/Hz \right] \tag{B.29}$$

where  $\Delta f$  denotes the offset frequency and  $f_{\rm osc}$  the oscillator frequency.

Based on Rover's specifications of -134 dBc/Hz at 1 MHz offset at a frequency of 1 GHz [4], one finds  $N_{\text{osc}} = -194 \text{ dBc/Hz}$ .

The power of an oscillator can be estimated using a Figure-of-Merit (FoM) defined as

$$\operatorname{FoM}_{\operatorname{osc}} = 10 \log \left( N_{\operatorname{osc}} \frac{P_{\operatorname{osc}}}{1 \operatorname{mW}} \right) \, [\mathrm{dBc/Hz}]$$
 (B.30)

For a ring-oscillator a good FoM is 160 dBc/Hz [136], while for an LC-oscillator 185 dBc/Hz can be achieved [65]. Using these FoMs and eq. (B.30), the power consumption of the ring-oscillator is 2.5 W, while the power consumption of the LC-oscillator is only 8 mW.

## **B.8** Input impedance of RF-frontend

The recurrence relation is

$$Z_{\rm in}[1] = 2R \ [\Omega]$$
  
$$Z_{\rm in}[n] = 2R \frac{R + Z_{\rm in}[n-1]}{3R + Z_{\rm in}[n-1]} \ [\Omega]$$

Evaluating this for several n gives the values of table B.1. From these values it can be observed that the numerator coefficient is always one more than the denominator coefficient. For the numerator p we postulate just by looking at the numbers that

$$p_n = 4p_{n-1} - 2$$

This is a simple first-order difference equation which can be solved using standard techniques, resulting in the solution (with  $p_1 = 2$  as initial condition)

$$p_n = \frac{4^n + 2}{3}$$

With the denominator coefficient q one less than the numerator coefficient p, one can simply subtract 1 from the solution for the numerator:

$$q_n = \frac{4^n - 1}{3}$$

The factors  $\frac{1}{3}$  cancel, yielding as general solution

$$Z_{\rm in}[n] = \frac{4^n + 2}{4^n - 1} R \ [\Omega]$$

All that needs to be done now is prove that this is indeed a correct solution, which will be done using mathematical induction. For n = 1 one finds  $Z_{in}[n = 1] = 2R$ , which is correct. Now assume the solution is correct for n = k. Filling in the solution for n = k + 1gives

$$Z_{in}[k+1] = 2R \frac{R + Z_{in}[k]}{3R + Z_{in}[k]}$$
  
=  $2R \frac{R + \frac{4^n + 2}{4^n - 1}R}{3R + \frac{4^n + 2}{4^n - 1}R}$   
=  $2R \frac{(4^n - 1)R + (4^n + 2)R}{(4^n - 1)3R + (4^n + 2)R}$   
=  $2R \frac{2 \cdot 4^n + 1}{4 \cdot 4^n - 1}$   
=  $R \frac{4^{n+1} + 2}{4^{n+1} - 1}$ 

which proves the solution is correct for all  $n \in \mathbb{N}$ .

# **B.9** Noise Figure of a Tayloe mixer

A Tayloe mixer is a mixer with a 25% duty cycle. Because the oscillator is a block wave, it contains many harmonics and therefore a lot of noise is folded to the Intermediate Frequency (IF), which increases the Noise Figure (NF).

The NF can be calculated by expanding the 25% duty cycle block wave s(t) into its Fourier components. Here the complex notation is chosen because it is more convenient.

$$s(t) = \sum_{n = -\infty}^{\infty} c_n e^{\frac{j2\pi nt}{T}}$$

where  $c_n$  can be calculated using

$$c_n = \frac{1}{T} = \int_{\langle T \rangle} s(t) e^{\frac{-j2\pi nt}{T}} \,\mathrm{d}t$$

Evaluating  $c_n$  gives

$$c_{n} = \frac{1}{T} \int_{0}^{\frac{T}{4}} e^{\frac{-j2\pi nt}{T}} dt = \frac{j}{2\pi n} \left( (-1)^{-\frac{n}{2}} - 1 \right)$$

$$= \begin{cases} \frac{1}{4} & \text{if } n = 0, \\ 0 & \text{if } n = 4k, \ k \in \mathbb{Z}, \\ \frac{1}{2\pi n} + \frac{1}{j2\pi n} & \text{if } n = 1 + 4k, \ k \in \mathbb{Z}, \\ \frac{1}{j\pi n} & \text{if } n = 2 + 4k, \ k \in \mathbb{Z}, \\ -\frac{1}{2\pi n} + \frac{1}{j2\pi n} & \text{if } n = 3 + 4k, \ k \in \mathbb{Z} \end{cases}$$
(B.31)



Figure B.1: Schematic for calculating total NF of cascaded noisy stages (reproduced from [44]).

For the NF one needs to calculate the noise factor as

$$F = \frac{\sum_{n=-\infty}^{\infty} |c_n|^2}{|c_{-1}|^2}$$
(B.32)

Using Parseval's theorem it can be seen that the numerator is simply the average power of the signal, which with a duty cycle of  $\frac{1}{4}$  is  $\frac{1}{4}$ .  $|c_{-1}|^2 = 1/2\pi^2$ , which means  $F = \pi^2/2$ , resulting in an NF of 6.9 dB, in accordance with [60].

When the circuit is balanced, all even harmonics are removed. For a duty cycle of 25% one removes all harmonics  $n = 2 + 4k, k \in \mathbb{Z}$ , because the other even harmonics are already zero. Evaluating the total power of the even harmonics gives

$$\sum_{k=-\infty}^{\infty} \left| c_{2+4k} \right|^2 = \frac{1}{8} \tag{B.33}$$

which is exactly half the total power of all harmonics. This means that in a balanced circuit the NF is decreased by 3 dB as compared to the unbalanced case, resulting in a NF of 3.9 dB.

## **B.10** Noise Figure of RF-frontend

The NF of the Tayloe-mixer is 6.9 dB if no balancing and I/Q-mixing is present [60]. The noise factor therefore is  $F \approx 4.898$ . This is defined for impedance-matched systems, but in the designed Radio Frequency (RF)-frontend the matching depends on the branch the mixer is connected to. Furthermore, the mixer is not the only part; there is a two-stage cascade of the resistor-network and the Tayloe mixer. This requires the use of a couple of formulas from [44] to derive the total NF. Some of the variables used are defined in fig. B.1.

The first equation is the well-known Friis equation [44, p. 45, eq. (2.107)]:

$$F_{\text{total}} = F_1 + \frac{F_2 - 1}{G_1} + \frac{F_3 - 1}{G_1 G_2} + \dots + \frac{F_n - 1}{\prod_{k=1}^{n-1} G_k}$$
(B.34)

where  $F_i$  is the noise factor with respect to the source impedance driving stage *i*, i.e. the output impedance of stage i - 1.  $G_i$  is the *available power gain* of stage *i* [44, p. 45, eq. (2.104)]:

$$G_{i} = \left(\frac{R_{\text{in}_{i}}}{R_{\text{out}_{i-1}} + R_{\text{in}_{i}}}\right)^{2} A_{v,i}^{2} \frac{R_{\text{out}_{i-1}}}{R_{\text{out}_{i}}}$$
(B.35)

with  $R_{\text{out}_0} = R_S$ , in the current design the antenna impedance.  $A_{v,i}$  is the voltage gain of stage i,  $R_{\text{in}_i}$  the input impedance of stage i and  $R_{\text{out}_i}$  the output impedance of stage i.

Comple	ex mult.	Compu	tations
Mult.	Add.	Mult.	Add.
3	3	$\frac{3}{2}LM - 5M + 8$	$\frac{7}{2}LM - 5M + 8$
4	2	$\bar{2LM} - 7M + 12$	$\overline{3}LM - 3M + 4$

Table B.2: Calculational complexity for an  $(M = 2^L)$ -point radix-2 complex FFT.

Next, a formula is needed to convert a given NF at one impedance to a NF at another impedance. This is given in [44, p. 201, eq. (6.36)]:

$$(F_A - 1)R_{S_A} = (F_B - 1)R_{S_B} \tag{B.36}$$

where  $F_A$  is the noise factor of the stage if driven by a source impedance  $R_{S_A}$ , and  $F_B$  the noise factor if driven by  $R_{S_B}$ .

The NF of the first stage (the resistor network) can be calculated using [44, p. 42, eq. (2.88)]:

$$F = 1 + \frac{R_S}{R_P} \tag{B.37}$$

where  $R_S$  is the source impedance and  $R_P$  the input impedance of the stage.

This design always has  $R_S = R_{S_A} = 50 \ \Omega$ . If the Tayloe mixer is connected to the first branch, eq. (5.1) gives  $R_P = R_{\text{in}_1} = 70 \ \Omega$ . The voltage gain of the resistor network is  $A_{v_1} = 1$  because the input is the output in this case. The output impedance follows from eq. (5.5) and is  $R_{\text{out}_1} \approx 29.17 \ \Omega$ . One finds  $G_1 \approx 0.583$ ,  $F_1 \approx 1.714$  and  $F_2 \approx 7.682$  and as a total noise factor  $F_{\text{total}} \approx 13.175$ . This is equal to NF<sub>total</sub> = 11.2 dB.

## B.11 Algorithmic complexity of complex FFT

Sorensen et al. [98] show a table where the required number of real additions and real multiplications are given for calculating an M-point complex FFT. They do this when calculating a complex multiplication using 4 real multiplications and 2 real additions, and when calculating a complex multiplication using 3 real multiplications and 3 real additions.<sup>3</sup> The table is reproduced for convenience in table B.2.

Given the fact that the number of calculations for both ways of calculating a complex multiplication contain a fixed number of real additions and multiplications, one can use the figures in table B.2 to calculate the number of complex multiplications for an  $(M = 2^L)$ -point radix-2 complex FFT. Denote  $x_1$  as the fixed number of real multiplications,  $x_2$  as the fixed number of real additions and y as the number of complex multiplications. The following system of equations then has to be solved:

$$x_1 + 3y = \frac{3}{2}LM - 5M + 8$$
  

$$x_1 + 4y = 2LM - 7M + 12$$
  

$$x_2 + 3y = \frac{7}{2}LM - 5M + 8$$
  

$$x_2 + 2y = 3LM - 3M + 4$$

of which the solution is

$$x_1 = M - 4$$
  $x_2 = 2LM - 4$   $y = \frac{1}{2}LM - 2M + 4$ 

<sup>&</sup>lt;sup>3</sup>Sorensen et al. claim a method exists to calculate a complex multiplication with three real multiplications and three real additions, but neither the calculation nor any other reference to this method has been found.
Appendix C

### Low Power VCO Idea

An oscillator design is proposed in chapter 4, but with a relatively high power consumption. A solution is to lower the phase noise requirement of the oscillator and reduce it through correlation, but because measurement time is increased fourfold if the noise is increased by 3 dB, it is desirable to fulfil the phase noise requirement with less power.

The required tuning range covers the 100 MHz to 6 GHz area, while the phase noise requirement derived in [4] is -134 dBc/Hz at an offset of 1 MHz with an oscillator frequency of 1 GHz. In section B.7 it is calculated that for this noise performance a ring-oscillator would require 2.5 W, while an LC-oscillator requires only 8 mW. These numbers are for the oscillator only, and do not include the power consumption required for the buffers. The high power consumption immediately rules out the use of only a ring-oscillator to generate the frequencies. The very limited tuneability of the LC-oscillator immediately rules out its standalone use. A solution is proposed here that uses the tunability of the ring-oscillator in combination with the superior phase noise of an LC-oscillator. This design sketch is far from worked out, and more analysis is needed to obtain a good understanding of the properties and feasability of the concept, which is left as future work.

The block-level design of the proposed solution is shown in fig. C.1. It consists of a High Frequency (HF) LC-oscillator, connected through a frequency divider to a Phase-Locked Loop (PLL), that is locked to a reference crystal resonator. The LC-oscillator is designed to meet the phase noise requirements of [4], but is not tunable. In this PLL the Voltage-Controlled Oscillator (VCO) is implemented using a ring-oscillator. The VCO can be rather noisy as will be explained further on. The detector and Low-Pass Filter (LPF) are standard elements in a PLL.

The PLL makes sure that the two inputs of the detector are in phase (and hence have the same frequency  $f_{\rm ref}$ ). This means  $f_{\rm out} = N f_{\rm ref} = \frac{N}{M} f_{\rm LC}$ , which provides us with a means to create a fraction of the frequency of the LC-oscillator. Depending on the range N and M can handle, one can pretty much set any desired frequency below  $f_{\rm LC}$ . Phase noise of the frequency dividers can be kept to a minimum by using re-clocking [71]. The input signal of the divider, which is not affected by the accumulated phase noise of the different stages in this divider, is also used as output of the divider. The lower frequency generated by the divider is used as 'enable' signal for the final stage, such that the output phase noise is only affected by the phase noise added in the last stage.

As a rule of thumb, the bandwidth of a PLL is usually set to roughly 10% of  $f_{\rm ref}$  [87]. Within this bandwidth, the LPF ideally filters all noise, because of the high-pass transfer characteristic to  $f_{\rm out}$ . The only remaining noise then is the noise from the LC-oscillator, which was within specifications. Outside of the bandwidth the filter does not reduce the noise; the noise level there comes from the RC-oscillator, which is not within specifications. A sketch of the output spectrum is shown in fig. C.2.

Any measurement done using the Spectrum Analyzer (SA) should not use frequencies



Figure C.1: Block-level schematic of the proposed oscillator design



Figure C.2: Output spectrum of the proposed oscillator design.

further away than  $f_{\rm ref}/10$ . Since  $f_{\rm ref} = f_{\rm out}/N$ , the measurement bandwidth is limited to  $f_{\rm ref}/(10N)$ . This means that if N = 10, one can only measure a 1 MHz bandwidth around 100 MHz within specifications. Therefore, the maximum relative measurement bandwidth available is 10% for N = 1.

This proposed solution combines the good phase noise of an LC-oscillator with the good tunability of an RC-oscillator, at the cost of a reduction in measurement bandwidth. Even when it does not meet the -134 dBc/Hz requirement, every improvement means less noise is added that needs to be correlated away.

Appendix D

## **Stochastic Processes**

In this appendix, a few basic statistic principles are briefly presented, as well as some signal and estimation theory. They should provide enough information to be able to understand the main text. Definitions provided here are much more formal than the main text, but this is only to avoid any confusion. More information can be found in any book on statistics and signal theory, such as [27, 28, 32, 42].

Deterministic signals are predictable, e.g. one can exactly predict the future values. An example is  $\sin(x)$ . *Stochastic*, or *random*, variables on the other hand, are not exactly predictable. An example is the number thrown with a die. The *sample space* is the set of all possible outcomes of the experiment.

**Definition D.1** (Stochastic variable). Let S be the sample space of an experiment and V be some value space. The function  $X : S \to V$  that assigns to each outcome  $s \in S$  a value  $X(s) = v, v \in V$ , is called a stochastic variable.

If  $V \subset \mathbb{R}$ , X is numeric, otherwise it is categorical. If the number of elements in V is finite or countably infinite, X is discrete. Hereafter, it is assumed that X is numeric.

In the case X is the number thrown with a die, the sample space is

 $\{1 \text{ eye thrown}, \ldots, 6 \text{ eyes thrown}\}$ 

and the value space is  $\{1, 2, 3, 4, 5, 6\}$  (with X providing the obvious mapping).

The cumulative distribution function  $F_X(x)$  gives the probability  $P(X \le x)$ . Similarly, the probability density function  $f_X(x)$  gives the probability P(X = x), which in the case of discrete variables amounts to a series of Dirac pulses  $(\delta(x))$ .  $f_X(x)$  can be regarded as  $\frac{dF_X(x)}{dx}$ , where the Dirac impulse is considered the derivative of a step.

The expected value E[X], or expectation, of a stochastic variable X, is the probabilityweighted average of all outcomes.

**Definition D.2** (Expectation). Let X be a discrete stochastic variable and Y a continuous stochastic variable. Then

$$E[X] = \sum_{x \in S_X} x P(X = x)$$

(with  $S_X$  the sample space of X) is the expectation of X, provided the summation is absolutely convergent, and

$$E[Y] = \int_{-\infty}^{+\infty} y f_Y(y) \, \mathrm{d}y$$

is the expectation of Y, provided the integral is absolutely convergent.

Another commonly used word for expectation is *mean*. The mean of X is then denoted as  $\mu_X$ . Other commonly used ways of writing the expectation E[X] are  $\overline{X}$  and  $\langle X \rangle$ . Note that in this thesis  $\overline{X}$  does not denote the mean of X, but the complex conjugate of X. Although different variables can have the same mean, they may exhibit completely different deviations from this mean. One way of characterizing this is using the *variance*, which is defined as the average quadratic deviation from the mean.

**Definition D.3** (Variance). Let X be a complex stochastic variable. The variance of X, var (X), is defined as var  $(X) = E\left[|X - E[X]|^2\right]$ .

If X is numeric, rewriting using the definition of E[X] gives var  $(X) = E[X^2] - E^2[X]$ .

A stochastic process (or random process) is a process which value at each instant of time is a random variable. It can therefore be considered as a collection of random variables. These random variables may have any relation with respect to each other, but they all have the same domain S and image V.

**Definition D.4** (Stochastic process). Let  $T \subset \mathbb{R}$  and  $X_t : S \to V$  be a random variable indexed with t. A stochastic process X is a collection  $\{X_t : t \in T\}$ .

In this thesis the convention is used that X[k] denotes (an element of) a discrete stochastic process, and X(t) (an element of) a continuous stochastic process, where it will be clear from the context whether the process as a whole or an element of the process is meant.

The cumulative probability distribution function and probability density function are also defined for stochastic processes:

$$F_X(x_1, \dots, x_n; t_1, \dots, t_n) \stackrel{\Delta}{=} P(X(t_1) \le x_1, \dots, X(t_n) \le x_n)$$
$$f_X(x_1, \dots, x_n; t_1, \dots, t_n) \stackrel{\Delta}{=} \frac{\delta^n F_X(x_1, \dots, x_n; t_1, \dots, t_n)}{\delta x_1 \cdots \delta x_n}$$

One can imagine that if process characteristics do not change in time, absolute time is not important. Difference in time however can still be important. This time-independence can be captured with the concept of *stationarity*.

**Definition D.5** (First-order stationary process). Let  $T \subset \mathbb{R}$  and X be a stochastic process. If  $\forall \tau \in T$  the equality  $f_X(x_1; t_1) = f_X(x_1; t_1 + \tau)$  holds, then X is a first-order stationary process.

A consequence of this property is that E[X(t)] is a constant.

**Definition D.6** (*N*th-order stationary process). Let  $T \subset \mathbb{R}$  and X be a stochastic process. If the equality  $f_X(x_1, \ldots, x_n; t_1, \ldots, t_n) = f_X(x_1, \ldots, x_n; t_1 + \tau, \ldots, t_n + \tau)$  holds  $\forall \tau \in T$ , then X is an Nth-order stationary process.

Note that (N + 1)-th order stationarity implies N-th order stationarity. For a secondorder stationary process it follows that  $f_X$  is only a function of the time difference  $t_2 - t_1$ , usually denoted as  $\tau$ .

Because even second-order stationarity places severe constraints on a process, and one would like to treat as many different processes at once, it is useful to introduce the concept of wide-sense stationary (wss) processes.

**Definition D.7** (Wide-sense stationary process). Let X(t) be a stochastic process. If E[X(t)] is a constant, and  $E[X(t)X(t+\tau)]$  is independent of t, then X(t) is a was process.

Note that all second-order stationary processes are wss, but the converse is not true. Similarly, one can define two processes to be jointly wss.

**Definition D.8** (Jointly wide-sense stationary processes). Let X(t) and Y(t) be two wss processes. If  $E[X(t)Y(t+\tau)]$  is independent of t, then X(t) and Y(t) are jointly wss processes.

A stochastic process has a(n) (in)finite amount of possible values at a certain point of time. These possible values together are called an *ensemble*. The expected value E[X] therefore is an *ensemble average*. Because only one realization of such an ensemble can be measured, the expectation cannot be measured.

However, if one assumes that the process is was and the statistical properties of one realization over time are equal to the ensemble average of the stochastic process at one instant of time, it can be measured. A process that has these two properties is called *ergodic*. The time average of a stochastic process is defined in the same way as expectation.

**Definition D.9** (Time average of a stochastic process). The time average A[X(t)] of a stochastic process X(t) is written as A[X(t)] and is defined as

$$A[X(t)] \stackrel{\triangle}{=} \lim_{T \to \infty} \frac{1}{2T} \int_{-T}^{T} x(t) dt$$

Because taking a time average involves only one realization of the stochastic process, A[X(t)] is itself a random variable. For ergodic processes however it is assumed that A[x(t)] converges almost surely (i.e. with probability 1) to E[A[X(t)]]. Hence one may write A[X(t)] instead of E[A[X(t)]].

The definition of time average allows the definition of an ergodic process.

**Definition D.10** (Ergodic process). Let X(t) be a wss process. Then if

$$A[X(t)] = E[X(t)]$$
  

$$A[X(t)X(t+\tau)] = E[X(t)X(t+\tau)]$$

X(t) is an ergodic process.

It must be noted that it is often impossible to prove that a process is ergodic, but for simplicity of measurement this is often assumed.

Similarly, one can define two processes to be jointly ergodic.

**Definition D.11** (Jointly ergodic processes). Let X(t) and Y(t) be two ergodic processes. If  $A[X(t)Y(t+\tau)] = E[X(t)Y(t+\tau)]$ , then X(t) and Y(t) are jointly ergodic processes.

Properties of stochastic processes need to be *estimated* by measurements. Functions or rules that estimate parameters based on an observation are called *estimators*. Estimators have several important properties. Because these functions act on stochastic processes, they are stochastic themselves, and therefore have an expectation and variance.

Ideally, an estimator has an expectation equal to the property it tries to determine, which leads to the definition of bias:

**Definition D.12** (Bias). Let  $\Theta$  be an estimator for unknown  $\theta$ . If  $E[\Theta] = \theta$ , then  $\Theta$  is *unbiased*, otherwise it is *biased* with *bias*  $B[\Theta] \stackrel{\triangle}{=} E[\Theta] - \theta$ . If N denotes the number of samples or the observation time, then if  $\lim_{N\to\infty} E[\Theta] = \theta$ ,  $\Theta$  is *asymptotically unbiased*.

Ideally, the variance of an estimator is zero. In general, the variance of an estimator is a function of the number of samples or observation time

**Definition D.13** (Consistency). Let  $\Theta$  be an estimator for unknown  $\theta$ . If  $\lim_{N\to\infty} \operatorname{var}(\Theta) = 0$ ,  $\Theta$  is *consistent*, otherwise it is *inconsistent*.

An often-used way to determine the quality of an estimator is the mean-squared error (mse), which uses both the bias and the variance.

**Definition D.14** (Mean-squared error). Let  $\Theta$  be an estimator for unknown  $\theta$ . Then  $\operatorname{mse}(\Theta) \stackrel{\triangle}{=} E\left[(\Theta - \theta)^2\right]$  is the mse of  $\Theta$ .

This can also be rewritten as  $\operatorname{mse}(\Theta) = (B[\Theta])^2 + \operatorname{var}(\Theta).$ 

# Bibliography

- [1] R.N. Bracewell. The Fourier Transform and its Applications. McGraw-Hill, 2nd edition, 1986.
- [2] A. Maußner. Konjunkturtheorie. Springer, 1994.
- [3] Agilent Technologies. Spectrum analysis basics. Application Note 150, 2004.
- [4] K.C. Rovers. Front-end research for a low-cost spectrum analyser. Master's thesis, Universiteit Twente, jun 2006.
- [5] G.E. Moore. No exponential is forever: But "forever" can be delayed! In *ISSCC*, volume 1, pages 20-23, 2003.
- [6] R. Nair. Integrity analysis and management in nanoscale SoC/SiP. In SoC, Tampere, Finland, nov 2007.
- [7] D. Ernst, N.S. Kim, S. Das, S. Pant, R. Rao, T. Pham, C. Ziesler, D. Blaauw, T. Austin, K. Flautner, and T. Mudge. Razor: A low-power pipeline based on circuit-level timing speculation. In *MICRO*, pages 7–18, dec 2003.
- [8] J.O. Mainardi, A.A.S. Júnior, L. Carro, and A.A. Susin. A comparison of totally digital ADCs for SOCs. In *ISCAS*, volume 1, pages 641–644, may 2004.
- [9] E.A. Vittoz. Low-power design: Ways to approach the limits. In *ISSCC*, pages 14–18, San Francisco, CA, feb 1994.
- [10] A.P. Jose, K.A. Jenkins, and S.K. Reynolds. Onchip spectrum analyzer for analog built-in self test. In *Proceedings of the 23rd IEEE VLSI Symposium*, 2005.
- [11] M.G. Méndez-Rivera, A. Valdes-Garcia, J. Silva-Martinez, and E. Sánchez-Sinencio. An on-chip spectrum analyzer for analog built-in testing. J. *Electron. Test.*, 21:205–219, 2005.
- [12] M.A. Domínguez, J.L. Ausín, and J.F. Duque-Carrillo. A 1-MHz area-efficient on-chip spectrum analyzer for analog testing. J. Electron. Test., 22: 437–448, 2006.
- [13] K.A. Jenkins and S. Polonsky. Integrated CMOS spectrum analyzer for on-chip diagnostics using digital autocorrelation of coarsely quantized signals. United States Patent 7,218,091, may 2007.

- [14] K.A. Jenkins, A.P. Jose, and S.K. Reynolds. Integrated spectrum analyzer circuits and methods for providing on-chip diagnostics. United States Patent 7,116,092, jan 2007.
- [15] Q. Zhang, F.W. Hoeksema, A.B.J. Kokkeler, and G.J.M. Smit. Mobile Multimedia: Communication Engineering Perspective, chapter 5: Towards Cognitive Radio for Emergency Networks, pages 75–100. Nova Publishers, 2006.
- [16] J. Yang, R.W. Brodersen, and D. Tse. Addressing the dynamic range problem in cognitive radios. In *ICC*, pages 5183–5188, Glasgow, Scotland, jun 2007.
- [17] V.J. Arkesteijn. Analog Front-Ends for Software-Defined Radio Receivers. PhD thesis, Universiteit Twente, 2007.
- [18] P.B. Kenington and L. Astier. Power consumption of A/D converters for software radio applications. *IEEE Trans. Veh. Technol.*, 49(2):643–650, mar 2000.
- [19] C. Rauscher. Fundamentals of Spectrum Analysis. Rohde & Schwarz GmbH, first edition, 2001.
- [20] F. Behbahani, Y. Kishigami, J. Leete, and A.A. Abidi. CMOS mixers and polyphase filters for large image rejection. *IEEE J. Solid-State Circuits*, 36(6):873–887, jun 2001.
- [21] J. Crols and M.S.J. Steyaert. A single-chip 900 MHz CMOS receiver front-end with a high performance low-IF topology. *IEEE J. Solid-State Circuits*, 30(12):1483-1492, dec 1995.
- [22] T. Manku, C. Snyder, M. Ting, F. Ling, J. Khajehpour, B. Kung, and L. Wong. Dual mixer downconversion architecture using complex mixing signals: Enabling solutions for software defined radios. In *CICC*, pages 227–234, 2002.
- [23] B. Razavi. Design considerations for directconversion receivers. *IEEE Trans. Circuits Syst. II, Analog Digit. Signal Process.*, 44(6):428–435, jun 1997.
- [24] J.M. Wing. Power measurement using dual spectrum analyzers. UK Patent Application, nov 2006. Company: Agilent Technologies.
- [25] S.S. Soliman and M.D. Srinath. Continuous and Discrete Signals and Systems. Prentice Hall Information and System Sciences Series. Prentice Hall, Upper Saddle River, NJ, second edition, 1998.

#### BIBLIOGRAPHY

- [26] M. Sampietro, L.G. Fasoli, and G. Ferrari. Spectrum analyzer with noise reduction by crosscorrelation technique on two channels. *Rev. Sci. Instrum.*, 70(5):2520–2525, may 1999.
- [27] W.C. van Etten. Introduction to Random Signals and Noise. John Wiley & Sons, 2005.
- [28] J.G. Proakis, C.M. Rader, F. Ling, C.L. Nikias, M. Moonen, and I.K. Proudler. Algorithms for Statistical Signal Processing. Prentice Hall, 2002.
- [29] A.M. Yaglom. Einstein's 1914 paper on the theory of irregularly fluctuating series of observations. *IEEE ASSP Mag.*, 4(4):7–11, oct 1987.
- [30] W.A. Gardner. Introduction to Einstein's contribution to time-series analysis. *IEEE ASSP Mag.*, pages 4–5, oct 1987.
- [31] J. Park, T. Song, J. Hur, S.M. Lee, J. Choi, K. Kim, J. Lee, K. Lim, C-H. Lee, H. Kim, and J. Laskar. A fully-integrated UHF receiver with multi-resolution spectrum-sensing (MRSS) functionality for IEEE 802.22 cognitive-radio applications. In *ISSCC*, pages 526–528, feb 2008.
- [32] G.M. Jenkins and D.G. Watts. Spectral Analysis and its Applications. Holden-Day, 1968.
- [33] E. Bagarinao and C. Saloma. Frequency analysis with Hopfield encoding neurons. *Physical Review* E, 54(5):5516-5521, nov 1996.
- [34] J.F. Bouchard, C. Zhu, and F.W. Paul. A neural network spectrum analyzer. *Mechatronics*, 5(6): 603–622, 1995.
- [35] D. Brogioli and A. Vailati. Real-time wavelettransform spectrum analyzer for the investigation of 1/f<sup>α</sup> noise. Rev. Sci. Instrum., 74(4):2583– 2592, apr 2003.
- [36] D. Mirri, G. Iuculano, G. Pasini, F. Filicori, and L. Peretto. A broad-band power spectrum analyzer based on twin-channel delayed sampling. *IEEE Trans. Instrum. Meas.*, 47(5):1346–1354, oct 1998.
- [37] G. Pasini, D. Mirri, G. Iuculano, and F. Filicori. Implementation and performance evaluation of a broad-band power spectrum analyzer. *IEEE Trans. Instrum. Meas.*, 50(4):1024–1029, aug 2001.
- [38] L. Peretto, G. Pasini, and C. Muscas. Signal spectrum analysis and period estimation by using delayed signal sampling. *IEEE Trans. Instrum. Meas.*, 50(4):920–925, aug 2001.
- [39] J.A.H. Harmsen. Design of a wideband jammer detector. Master's thesis, Universiteit Twente, aug 2003.
- [40] Rohde & Schwarz GmbH. FSUP Quick Start Guide, jan 2008.
- [41] P.D. Welch. The use of fast fourier transform for the estimation of power spectra: A method based on time averaging over short, modified periodograms. *IEEE Trans. Audio Electroacoust.*, 15(2):70–73, jun 1967.
- [42] A.M. Yaglom. Correlation Theory of Stationary and Related Random Functions, volume I: Basic Results of Springer Series in Statistics. Springer-Verlag, 1987.
- [43] A.B.J. Kokkeler. Analog-Digital Codesign using Coarse Quantization. PhD thesis, Universiteit Twente, apr 2005.

- [44] B. Razavi. *RF Microelectronics*. Prentice Hall Communications Engineering and Emerging Technologies Series. Prentice Hall, 19 edition, dec 2006.
- [45] M. Macucci and B. Pellegrini. Very sensitive measurement method of electron device current noise. *IEEE Trans. Instrum. Meas.*, 40(1):7–12, feb 1991.
- [46] G. Ferrari and M. Sampietro. Material and device characterization using a correlation spectrum analyzer. *Mat. Sci. Semicon. Proc.*, 4:133–136, 2001.
- [47] G. Ferrari and M. Sampietro. Correlation spectrum analyzer for direct measurement of device current noise. *Rev. Sci. Instrum.*, 73(7):2717– 2723, jul 2002.
- [48] B. Le, T.W. Rondeau, J.H. Reed, and C.W. Bostian. Analog-to-digital converters: A review of the past, present and future. *IEEE Signal Process. Mag.*, pages 69–77, nov 2005.
- [49] R.H. Walden. Analog-to-digital converter survey and analysis. *IEEE J. Sel. Areas Commun.*, 17 (4):539–550, April 1999.
- [50] J.H. van Vleck and D. Middleton. The spectrum of clipped noise. Proc. IEEE, 54(1):2–19, jan 1966.
- [51] N.M. Blachman. The intermodulation and distortion due to quantization of sinusoids. *IEEE Trans. Acoust., Speech, Signal Process.*, 33(6): 1417–1426, dec 1985.
- [52] H. Pan and A.A. Abidi. Spectral spurs due to quantization in Nyquist ADCs. *IEEE Trans. Circuits Syst. I, Reg. Papers*, 51(8):1422–1439, aug 2004.
- [54] M. Abramowitz and I.A. Stegun. Handbook of Mathematical Functions With Formulas, Graphs and Mathematical Tables. Dover Publications, June 1964.
- [55] M.F. Wagdy. Effect of various dither forms on quantization errors of ideal A/D converters. *IEEE Trans. Instrum. Meas.*, 38(4):850–855, August 1989.
- [56] A.B.J. Kokkeler and A.W. Gunst. Modeling correlation of quantized noise and periodic signals. *IEEE Signal Process. Lett.*, 11(10):802–805, oct 2004.
- [57] D. Bellan, A. Brandolini, and A. Gandelli. ADC nonlinearities and harmonic distortion in FFT test. In *IMTC*, pages 1233–1238, St. Paul, MN, may 1998.
- [58] H.K. Chen, D.C. Chang, Y.Z. Juang, and S.S. Lu. A compact wideband CMOS low-noise amplifier using shunt resistive-feedback and series inductive-peaking techniques. *IEEE Microw. Wireless Compon. Lett.*, 17(8):616–618, aug 2007.
- [59] R.C. Liu, C.S. Lin, K.L. Deng, and H. Wang. A 0.5–14-GHz 10.6-dB CMOS cascode distributed amplifier. In Symposium on VLSI Circuits Digest, volume 17, pages 139–140, jun 2003.
- [60] M. Soer. Analysis and comparison of switch-based frequency converters. Master's thesis, Universiteit Twente, sep 2007.

- [61] N.A. Moseley, E.A.M. Klumperink, and B. Nauta. A two-stage approach to harmonic rejection mixing using blind interference cancellation. *IEEE Trans. Circuits Syst. II, Exp. Briefs*, 55(10):966– 970, oct 2008.
- [62] E. Mensink, E.A.M. Klumperink, and B. Nauta. Distortion cancellation by polyphase multipath circuits. *IEEE Trans. Circuits Syst. I, Reg. Papers*, 52(9):1785–1794, sep 2005.
- [63] R. Shrestha, E.A.M. Klumperink, E. Mensink, G.J.M. Wienk, and B. Nauta. A polyphase multipath technique for software-defined radio transmitters. *IEEE J. Solid-State Circuits*, 41(12): 2681–2692, dec 2006.
- [64] Agilent Technologies. Fundamentals of quartz oscillators. Application Note 200–2, 1997.
- [65] F. Leong. Design of an oscillator for satellite reception. Master's thesis, Universiteit Twente, sep 2007.
- [66] P. Kinget, B. Soltanian, S. Xu, S. Yu, and F. Zhang. Advanced design techniques for integrated voltage controlled LC oscillators. In *CICC*, sep 2007.
- [67] T. Choi, H. Lee, L.P.B. Katehi, and S. Mohammadi. A low phase noise 10 GHz VCO in  $0.18 \mu$ m CMOS process. In *ECWT*, pages 273–276, oct 2005.
- [68] H.-M. Hsu and J.-Z. Chang. Mutual coupling of on-chip inductors in CMOS technology. J. Micromech. Microeng., 18(3):1–5, mar 2008.
- [69] R. Nonis, E. Palumbo, P. Palestri, and L. Selmi. A design methodology for MOS current-mode logic frequency dividers. *IEEE Trans. Circuits Syst. I, Reg. Papers*, 54(2):245–254, feb 2007.
- [70] C. Kromer, G. von Büren, G. Sialm, T. Morf, F. Ellinger, and H. Jäckel. A 40-GHz static frequency divider with quadrature outputs in 80-nm CMOS. *IEEE Microw. Wireless Compon. Lett.*, 16(10):564-566, oct 2006.
- [71] S. Levantino, L. Romanò, S. Pellerano, C. Samori, and A.L. Lacaita. Phase noise in digital frequency dividers. *IEEE J. Solid-State Circuits*, 39(5): 775–784, may 2004.
- [72] M. Arora. Clock dividers made easy. In SNUG, Boston, MA, USA, 2002.
- [73] X. Gao, E.A.M. Klumperink, and B. Nauta. Advantages of shift registers over DLLs for flexible low jitter multiphase clock generation. *IEEE Trans. Circuits Syst. II, Exp. Briefs*, 55(3):244– 248, mar 2008.
- B. Murmann. ADC performance survey 1997– 2007. http://www.stanford.edu/ murmann/adcsurvey.html, sep 2008.
- [75] S.-C. Lee, Y.-D. Jeon, K.-D. Kim, J.-K. Kwon, J. Kim, J.-W. Moon, and W. Lee. A 10b 205MS/s 1mm<sup>2</sup> 90nm CMOS pipeline ADC for flat-panel display applications. In *ISSCC*, pages 458–460, 2007.
- [76] S.M. Louwsma, A.J.M. van Tuijl, M. Vertregt, and B. Nauta. A 1.35 GS/s, 10b, 175 mW timeinterleaved AD converter in 0.13 μm CMOS. In *IEEE Symposium on VLSI Circuits*, pages 62– 63, jun 2007.

- [77] C. Aldea, S. Celma, and A. Otin. A 62 dB dynamic range sixth-order band pass filter with 100– 175 MHz tuning range. In *Proceedings of the 29th ESSCIRC*, pages 437–440, sep 2003.
- [78] N.A. Moseley, E.A.M. Klumperink, and B. Nauta. A spectrum sensing technique for cognitive radios in the presence of harmonic images. In *DYSPAN*, Chicago, IL, oct 2008.
- [79] C. Cordeiro, K. Challapali, D. Birru, and S. Shankar N. IEEE 802.22: An introduction to the first wireless standard based on cognitive radios. J. Commun., 1(1):38–47, apr 2006.
- [80] B. Razavi. A 60 GHz direct-conversion CMOS receiver. In *ISSCC*, 2005.
- [81] The Boeing Company. Signal analyzer systems. English Patent 1,062,966, mar 1967.
- [82] T.S. Lande, T.G. Constandinou, A. Burdett, and C. Toumazou. Running cross-correlation using bitstream processing. *Electron. Lett.*, 43(22): 1181–1182, 2007.
- [83] W. Huang and M. Ismail. A CMOS wideband LNA for DCS1800 PCS1900 and WCDMA. In MWSCAS, volume 3, pages 1235–1238, dec 2003.
- [84] R. Molavi, S. Mirabbasi, and M. Hashemi. A wideband CMOS LNA design approach. In *ISCAS*, volume 5, pages 5107–5110, may 2005.
- [85] T. Ström and S. Signell. Analysis of periodically switched linear circuits. *IEEE Trans. Circuits* Syst., 24(10):531–541, oct 1977.
- [86] T.W. Brown, T.S. Fiez, and M. Hakkarainen. Prediction and characterization of frequency dependent MOS switch linearity and the design implications. In *CICC*, pages 237–240, 2006.
- [87] B. Razavi. Design of Analog CMOS Integrated Circuits. MCGraw-Hill, 2001.
- [88] Hameg. HM5014–2 Spectrum Analyzer Datasheet, nov 2006.
- [89] Tektronix. RSA2200A Series Datasheet, feb 2006.
- [90] Rohde & Schwarz GmbH. R&S FSP Spectrum Analyzer Datasheet, may 2008.
- [91] Agilent Technologies. Agilent PSA Series Spectrum Analyzers Datasheet, aug 2008.
- [92] Agilent Technologies. Agilent N9340A Handheld Spectrum Analyzer Technical Overview, feb 2007.
- [93] SED Systems. SED Decimator datasheet, 2008.
- [94] Metrix. MTX 1050 PC Spectrum Analyzer Datasheet, 2008.
- [95] C.E. Rehorn and N.S. Barker. A miniaturized lowcost 60–1000-MHz PCB spectrum analyzer. *IEEE Trans. Instrum. Meas.*, 57(1):205–212, jan 2008.
- [96] Intel Corporation. Using streaming SIMD extensions 3 in algorithms with complex arithmetic, jan 2004.
- [97] E.W. Weisstein. Complex multiplication. From MathWorld-A Wolfram Web Resource, http://mathworld.wolfram.com/ ComplexMultiplication.html, 2008.
- [98] H.V. Sorensen, D.L. Jones, M.T. Heideman, and C.S. Burrus. Real-valued fast fourier transform algorithms. *IEEE Trans. Acoust., Speech, Signal Process.*, 35(6):849–863, jun 1987.
- [99] J.D. Bunton. SKA correlator advances. *Exp. Astron.*, 17(1–3):251–259, jun 2004.

#### [100] N.S. Kim, T. Austin, D. Blaauw, T. Mudge, K. Flautner, J.S. Hu, M.J. Irwin, M. Kandemir, and V. Narayanan. Leakage current: Moor's Law meets static power. *Computer*, 36(12):68–75, dec 2003.

- [101] D. Helms, E. Schmidt, and W Nebel. Integrated Circuit and System Design, volume 3254 of Lecture Notes in Computer Science, chapter Leakage in CMOS Circuits – An Introduction, pages 17–35. Springer, aug 2004.
- [102] F.N. Najm. A survey of power estimation techniques in VLSI circuits. *IEEE Trans. VLSI Syst.*, 2(4):446–455, dec 1994.
- [103] S.T. Oskuii, P.G. Kjeldsberg, and O. Gustafsson. Transition-activity aware design of reductionstages for parallel multipliers. In *GLSVLSI*, pages 120–125, Stresa-Lago Maggiore, Italy, 2007.
- [104] C. Svensson and A. Alvandpour. Low power and low voltage CMOS digital circuit techniques. In *ISLPED*, pages 7–10, Monterey, CA, 1998. ACM.
- [105] S. Bhanja and N. Ranganathan. Switching activity estimation of VLSI circuits using bayesian networks. *IEEE Trans. VLSI Syst.*, 11(4):558–567, aug 2003.
- [106] H. Mangassarian, A. Veneris, S. Safarpour, F.N. Najm, and M.S. Abadir. Maximum circuit activity estimation using pseudo-Boolean satisfiability. In *DATE*, pages 1–6, apr 2007.
- [107] M.A.T. Sanduleanu and A.J.M. van Tuijl. Power Trade-Offs and Low-Power in Analog CMOS ICs, volume 662 of The Springer International Series in Engineering and Computer Science, chapter Power Considerations in Sub-Micron Digital CMOS, pages 9–29. Springer Netherlands, 2003.
- [108] P.M. Heysters. Coarse-Grained Reconfigurable Processors. PhD thesis, Universiteit Twente, sep 2004.
- [109] A. Wang. An Ultra-Low Voltage FFT Processor Using Energy-Aware Techniques. PhD thesis, Massachusetts Institute of Technology, dec 2003.
- [110] P.M. Heysters, G.J.M. Smit, and E. Molenkamp. Energy-efficiency of the Montium reconfigurable tile processor. In *ERSA*, pages 38–44, Las Vegas, NV, 2004. CSREA Press.
- [111] T. Bijlsma, P.T. Wolkotte, and G.J.M. Smit. An optimal architecture for a DDC. In *IPDPS*, pages 192–200, Los Alamitos, CA, apr 2006. IEEE Computer Society.
- [112] L.T. Smit, G.K. Rauwerda, A. Molderink, P.T. Wolkotte, and G.J.M. Smit. Implementation of a 2-D 8x8 IDCT on the reconfigurable montium core. In *FPL*, pages 562-566, Amsterdam, NL, aug 2007. IEEE Computer Society Press.
- [113] G.K. Rauwerda, G.J.M. Smit, and W. Brugger. Implementing an adaptive viterbi algorithm in coarse-grained reconfigurable hardware. In *ERSA*, pages 62–68, Las Vegas, NV, jun 2005. ISBN 1-932415-74-2.
- [114] Recore Systems. Montium TP design specification. internal specification document, version 01.09.03, confidential, 2008.
- [115] Recore Systems. Montium version 2.0 user guide, oct 2008.
- [116] H. Hassler and N. Takagi. Function evaluation by table look-up and addition. In ARITH, pages 10–16, jul 1995.
- [117] T. Furuyama. Trends and challenges of large scale embedded memories. In ESSCIRC, pages 449– 456, 2004.

- [118] H. Pilo, C. Barwin, G. Braceras, C. Browning, S. Lamphier, and F. Towler. An SRAM design in 65-nm technology node featuring read and writeassist circuits to expand operating voltage. *IEEE J. Solid-State Circuits*, 42(4):813–819, apr 2007.
- [119] K. Itoh, S. Kimura, and T. Sakata. VLSI memory technology: Current status and future trends. In *ESSCIRC*, pages 3–10, sep 1999.
- [120] Texas Instruments. TMS320C6421 fixed-point digital signal processor datasheet, jun 2008.
- [121] A.H. Nuttall. Some windows with very good sidelobe behavior. *IEEE Trans. Acoust., Speech, Signal Process.*, 29(1):84–91, feb 1981.
- [122] A.A. Moulthrop and M.S. Muha. Accurate measurement of signals close to the noise floor on a spectrum analyzer. *IEEE Trans. Microw. The*ory Tech, 39(11):1882–1885, nov 1991.
- [123] Agilent Technologies. Spectrum analyzer measurements and noise. Application Note 1303, 2003.
- [124] G. Betta, M. D'Apuzzo, C. Liguori, and A. Pietrosanto. An intelligent FFT-analyzer. *IEEE Trans. Instrum. Meas.*, 47(5):1173–1179, oct 1998.
- [125] C.G. Gumas. Window-presum fft achieves highdynamic range, resolution. *Personal Engineering* & Instrumentation News, pages 58–64, 1997.
- [126] J. Lillington. Comparison of transient response of FFT PFT and polyphase DFT filter banks. White paper, RF Engines Ltd, apr 2008.
- [127] F. Lis and G. Pan. Speeding up the CORDIC algorithm with a DSP, sep 2008.
- [128] F.K. Bowers and R.J. Klingler. Quantization noise of correlation spectrometers. Astron. Astrophys. Sup., 15:373–380, jun 1974.
- [129] M.S. Keshner. 1 / f noise. Proc. IEEE, 70(3): 212–218, mar 1982.
- [130] A.P. van der Wel, E.A.M. Klumperink, S.L.J. Gierkink, R.F. Wassenaar, and H. Wallinga. MOSFET 1/f noise measurements under switched bias conditions. *IEEE Electron Device Lett.*, 21 (1):43–46, jan 2000.
- [131] E.A.M. Klumperink, S.L.J. Gierkink, A.P. van der Wel, and B. Nauta. Reducing MOSFET 1/f noise and power consumption by switched biasing. *IEEE J. Solid-State Circuits*, 35(7):994–1001, jul 2000.
- [132] M.R. Yuce and W. Liu. Alternative wideband front-end architectures for multi-standard software radios. In VTC, volume 3, pages 1968–1972, sep 2004.
- [133] J. Briaire and L.K.J. Vandamme. Uncertainty in gaussian noise generalized for cross-correlation spectra. J. Appl. Phys., 84:4370–4374, 1998.
- [134] E.W. Weisstein. Chi distribution. From MathWorld-A Wolfram Web Resource, http: //mathworld.wolfram.com/ChiDistribution.html, 2008.
- [135] T.H. Lee and A. Hajimiri. Oscillator phase noise: A tutorial. *IEEE J. Solid-State Circuits*, 35(3): 326–336, mar 2000.
- [136] M. Grözing, B. Philipp, and M. Berroth. CMOS ring oscillator with quadrature outputs and 100 MHz to 3.5 GHz tuning range. In *ESSCIRC*, sep 2003.

### BIBLIOGRAPHY