Full Spectrum Receiver Design

A case study of direct RF sampling

Master's Thesis
by

Joris Mosheuvel

Committee:
prof.dr.ir. M.J.G. Bekooij (CAES)
dr.ir. A.B.J. Kokkeler (CAES)
ir. B.H.J. Dekens (CAES)

University of Twente, Enschede, The Netherlands
January 13, 2014

This page is intentionally left blank.

UNIVERSITEIT TWENTE.

COMMIT/

Abstract

The ultimate Software-Defined Radio (SDR) front-end can concurrently capture and process any number of radio signals. In this thesis we present the implications and problems of the Full-Spectrum Receiver (FSR) which attempts to facilitate this functionality. We study the requirements for a practical receiver in order to relate the performance of an FSR to the classic radio receiver implementations. We identify the main difficulties and trade-offs in the architecture in both the signal reception and digital processing. We implement a prototype FSR on a Virtex-6 Field-Programmable Gate Array (FPGA) and use it to validate our statements.

Acknowledgements

For this research I thank my professor Marco Bekooij for the intensive support on this difficult topic. I thank professor André Kokkeler for reviewing my thesis and providing valuable feedback. I thank my supervisor Berend Dekens for the support with the hardware, software and a great deal of improvements to my work. My gratitude goes out to Koen Blom and Rinse Wester for providing me with the required information on signal processing to get me started on this topic. I especially thank Mark de Ruiter for his insight and valuable contributions in both the on- and off-topic discussions we had. Last, but not least, I thank Daphne Heij for her invaluable support for my work.

Contents

List of Figures

Acronyms

$\frac{f_s}{2}$ half the sample frequency.

f_c Carrier Frequency.

f_s sample frequency.

ADC Analog-to-Digital Converter.

ASIC Application Specific Integrated Circuit.

BPF Bandpass Filter.

BPR Band-Pass Receiver.

CIC Cascaded Integrator-Comb.

CORDIC COrdinate Rotation for DIgital Computers.

DC Direct Current.

DCR Direct Conversion Receiver.

DDC Digital Down Converter.

DDS Direct Digital Synthesizer.

DR Dynamic Range.

DVB Digital Video Broadcasting.

EEPROM Electrically Erasable Programmable ROM.

FBC Full-Band Capture.

FDA-Tool Filter Design and Analysis-Tool.

FE Front-End.

FFT Fast-Fourier Transform.

FIFO First-In First-Out.

FIR Finite Impulse Response.

FM Frequency Modulation.

FMC FPGA Mezzanine Card.

FOM Figure of Merit.

FPGA Field-Programmable Gate Array.

FSR Full-Spectrum Receiver.

GPS Global Positioning System.

GRC GNU-Radio Companion.

HF High-Frequency.

HPC High Pin-Count.

IF Intermediate Frequency.

IIR Infinite Impulse Response.

IMD Inter-Modulation Distortion.

IP intercept Point.

ISSCC International Solid-State Circuit Conference.

LFTX Low-Frequency Transmit.

LNA Low-Noise Amplifier.

LPF Low-Pass Filter.

LSB Least Significant Bit.

LUT Look-Up Table.

LVDS Low-Voltage Differential Signal.

NCO numerically Controlled Oscillator.

NF Noise Figure.

OOB Out Of Band.

PC Personal Computer.

RAM Random-Access Memory.

RDS Radio Data System.

RF Radio Frequency.

RMS Root Mean Square.

RX Receive.

S/H Sample-and-Hold.

SAW Surface Acoustic Wave.

SDR Software-Defined Radio.

SFDR Spurious-Free Dynamic-Range.

SHR Super-Heterodyne Receiver.

SJNR Signal to Jitter Noise Ratio.

SMC Sub-Miniature version C.

SNDR Signal-to-Noise and Distortion Ratio.

SNR Signal-to-Noise Ratio.

SPI Serial Peripheral Interface.

TX Transmit.

USB Universal Serial Bus.

USRP Universal Software Radio Peripheral.

VGA Variable-Gain Amplifier.

VSG Vector Signal Generator.

WB Wide-Band.

Chapter 1

Introduction

In this thesis we cover the design and implementation trade-offs of a Full-Spectrum Receiver (FSR). The FSR receives all signals from Direct Current (DC) voltage up to a certain maximum frequency without any analog filtering, to allow concurrent reception of multiple Radio Frequency (RF) signals using a single Front-End (FE). In this section we present the motivation for our research by presenting the FSR and compare it to other (conventional) receivers. We point out the main problems for the FSR architecture. We introduce the research questions to provide a general direction to our work . We present the contributions of our work in this report. Lastly we give the outlines of this thesis in order to give a general overview of the chapters.

1.1 Motivation

In this section we describe why we started our research on FSRs and why we are interested in this research. In order to do that we begin with the conventional architecture for receivers, the Super-Heterodyne Receiver (SHR). We address the shortcomings of the SHR and modify its structure to show what we try to overcome with the FSR. We finish this section with requirements of what would make a practical FSR.

1.1.1 Background

This research was inspired by the publication of RF-Engines [10] that presents a Wide-Band (WB) Digital Down Converter (DDC) for Field-Programmable Gate Arrays (FPGAs) with a sample frequency (f_s) of 2 [GHz]. The DDC performs the digital mixing and filtering for signals between DC and 1 [GHz]. Conventionally these steps are performed in the analog domain, specifically when sample frequencies in the GHz range (and above) are involved, because of the, otherwise, high data rates. It inspired us to look into the techniques of digital signal processing that are required for such a system.

Using digital signal processing instead of analog processing has many advantages, specifically the ability to dynamically select an arbitrary radio band. The DC to 1 [GHz] band, which is targeted in [10], covers the Frequency Modulation (FM) broadcasting band, Digital Video Broadcasting (DVB) signals

but also maritime frequencies, bands open for amateur radio enthusiasts and many others [11].

Because digital filters can be reconfigured to isolate any band up to half the sample frequency ($\frac{f_s}{2}$), the system becomes very flexible. This allows the reception of many frequency bands concurrently, using the same FE. This is called an FSR or Full-Band Capture (FBC) receiver. The FSR architecture looks like the ideal receiver architecture for Software-Defined Radio (SDR) which can receive, concurrently, every channel that the Analog-to-Digital Converter (ADC) digitizes [12].

1.1.2 Design of the Full-Spectrum Receiver

The SHR is currently the most used radio receiver for RF-reception. It is a receiver with excellent reception characteristics, but it limits the received signals to certain Carrier Frequency (f_c) by its mixers and the bandwidth because of analog filters. On top of that the SHR makes use of an Intermediate Frequency (IF) stage, increasing the system complexity. In the FSR we want to capture all signals from DC up to $\frac{f_s}{2}$ without pre-conditioning the bandwidth or f_c of the signal before it is digitized.

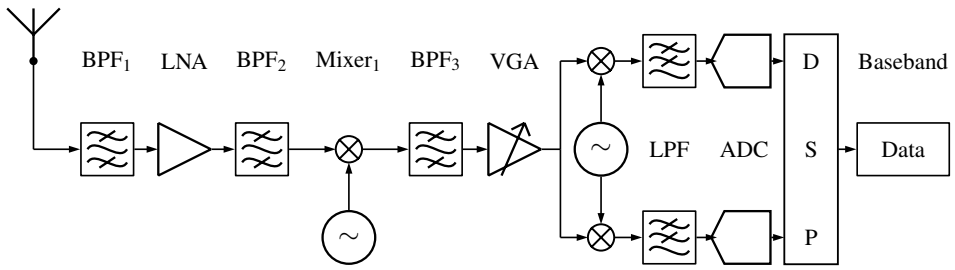


Figure 1.1: Super-heterodyne receiver according to [1].

In figure 1.1 we present the architecture of the SHR. The first Band-Pass Filter (BPF) performs the pre-selection of the signal to reject images that are far away in the frequency band. BPF_2 then performs another image-rejection after the amplification by the Low-Noise Amplifier (LNA). The first mixer ($\otimes$) performs the frequency shift from RF to some IF to have BPF_3 filter out the desired channel. The Variable-Gain Amplifier (VGA) performs an amplification so that the signal matches the full scale of the ADCs. The second mixer performs a translation from IF to baseband, after that the signal passes an Low-Pass Filter (LPF) and is finally digitized by the ADCs.

The (bandpass-)filters in the SHR are made of analog components and thus fixed to specific bands. They are required to condition the signals in the passband to match the ADC. Fixed filters mean the bandwidth of the receiver is limited. For our FSR design we do not want to be limited to specific bands so we remove the filters from the design; filtering can later be implemented in the

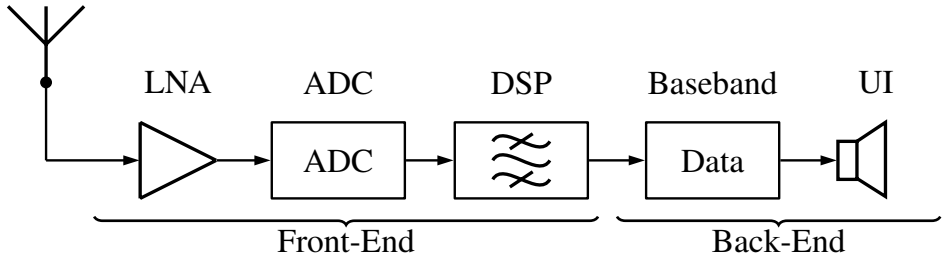


Figure 1.2: Full-spectrum receiver.

digital domain where they can be adjusted as desired. The mixing in the SHR is performed to shift the desired signal to the passband of the filters. Because we remove the filters in the FSR, mixing between stages becomes obsolete and we can remove them as well. The frequency translation to baseband remains necessary, but this can be performed in the digital part. Because the amplification of the VGA and the LNA is limited by the highest interferer, we have to determine its power. Then we can merge them to perform a fixed amplification and end up with the receiver architecture of figure 1.2.

1.1.3 Uses for Full-Spectrum Receivers

FSR technology is already successfully implemented in chips for DVB-receivers by NXP, Max Linear and Broadcom [13] [14] to replace the expensive tuners that are conventionally employed in those systems. When comparing the number of components in figures 1.1 and 1.2 we can clearly see that the FSR has much less (analog) blocks than the SHR. The reduced number of components makes the FSR circuitry cheaper, creating budget for a better ADC.

The DVB-FSRs are employed in home entertainment systems and receive digital television and radio on several channels at the same time, limited only by the number of filtering paths and demodulators. We found that this application is possible because in wired systems the signal strengths are kept quite similar. This means the ADC does not have to cover such a wide range of strengths which relaxes the requirements on the design of the ADC. In wireless RF-communication the antenna is the first part of the receiver that introduces a certain bandwidth, making its design crucial for receivers. Multichannel antennas are developed [15] as a (partial) solution to counter this property.

High-end FSRs could be interesting for intelligence agencies to monitor the ˜whole″spectrum and focus on specific radio signals that appear suspicious. Another application for FSRs may be as a radar front-end, where software filters can detect objects based on the received patterns. This would allow the radar to scan with a wide sweep-range with only a single ADC. When capturing the whole spectrum, the system is limited by its bandwidth (mainly

due to the sample frequency) and the number of demodulators for the different channels and transmission schemes (FM, QAM, PSK, etc.). Lastly the FSR lends itself perfectly for wideband communications because they are designed to receive a large bandwidth.

The FSR seems a very promising receiver architecture and very worthwhile to look into but there will be some considerations and trade-offs involved. The main challenge in its development will be for the ADC to receive both strong and weak signals at the same time.

1.2 Problem definition

Figure 1.3: Implementation of a conventional 8-channel receiver.

In figure 1.3 we see an implementation of a DVB cable multichannel receiver with 8 tuners placed next to each other. With an FSR these could all be replaced by a single chip; saving a huge amount of components. However, there are some drawbacks to this implementation that can not be overlooked. We have received some hardware to create and research an FSR implementation. In this section we define the minimum requirements that we think a practical wireless receiver should fulfill. With these specifications we can determine whether it is possible to create an FSR using the hardware that we have received. We will discuss the cost of the system and the trade-offs we have to make when realizing our FSR implementation.

1.2.1 Requirements

In this section we present the requirements for the signals that we want to receive with our FSR; this defines the specifications of our system. From [16] we find that the typical maximum signal strength for signals in the ether is defined as -13 [dBm]. Taking into account a small margin for this value we define the maximum input power 3 [dB] higher: $P_{in,max} = -10$ [dBm]. The smallest signal that can be received, defines the sensitivity of the receiver.

	$WiFi_{min}$	$WiFi_{max}$	Bluetooth	GPS	FM
Sensitivity [dBm]	-90	-65	-70	-130	-100
Maximum input signal [dBm]	-30	-30	-10	-90 (?)	-13
Dynamic Range [dB]	60	35	60	40	87

Table 1.1: Table showing the minimum, maximum and range of signal strengths for various standards.

In table 1.1 the signal strengths are given for WiFi (802.11a/b/g/n) minimum and maximum data rates [17], Classic Bluetooth [18], Global Positioning System (GPS) [19] and FM [20]. WiFi signals use a 22 [MHz] band, Bluetooth 1 [MHz], GPS 2 [MHz] and FM-radio $\sim$ 200 [kHz].

We choose to design an FM receiver as its modulation and demodulation is easily performed with the available tools. While FM can contain multiple signals for stereo and Radio Data System (RDS), we focus on a receiver for mono-audio as it is sufficient for our tests. A sensitivity of at least -90 [dBm] is a little higher than conventional FM receivers but it will allow us to receive commercial radio signals not too far from the broadcasting tower. The strongest signal that can be expected is -10 [dBm] and the ADC will have to facilitate for this signal power as well.

An LNA is commonly used in receivers to increase the sensitivity. Amplification only works if the signal power does not exceed the maximum input of the ADC. In conventional receivers interferers are filtered out early in the receiver chain by BPFs. With a VGA the Dynamic Range (DR) of the receiver can be increased, because the desired signal may be amplified according to its strength. If no filtering is applied, like in the FSR, the maximum possible amplification depends on the strongest signal, which might be an interferer. With the maximum signal power that the FSR should handle we defined its full-scale of the. With it, we can use an LNA to provide a constant amplification.

LNAs typically amplify between 15 and 25 [dB] with a Noise Figure (NF)[1] under 4 [dB]. The LNA will behave as a BPF because they can not uniformly amplify all frequencies . Typical values for our application range vary from

[1] NF is the expected decrease of the Signal-to-Noise Ratio (SNR) caused by a component.

100 [kHz] up to 2[GHz] but other bands are possible. Because there is a lot of well designed LNAs suitable for use in FSRs, we expect its bandwidth not to be an issue. The amplification and noise-figure however, will have a definite impact on the performance of the receiver. We defined that the FSR receiver should detect signals at -90 [dBm]. With the LNA amplification of 20 [dB] and an NF of 4 [dB], the receiver can detect signals as low as $-90+20+4 = -66$ [dBm]. The strongest signal will be 10 [dBm] at the input of the ADC resulting in a DR of 76 [dB].

Because we are designing an FSR, we should be able to receive at least two channels as a proof of concept. Our ADC has an f_s between 200 [MHz] and 5 [GHz] so we can capture signals up to 2.5 [GHz] which covers the IEEE-802.11b WiFi band (2.4 [GHz]).

Because of the bandwidth of to the LNA we will be limited to the band from 100 [kHz] to 1000 [MHz]. The flatness and roll-off of the LNA will define whether its possible to take a f_c outside this band.

1.2.2 Feasibility

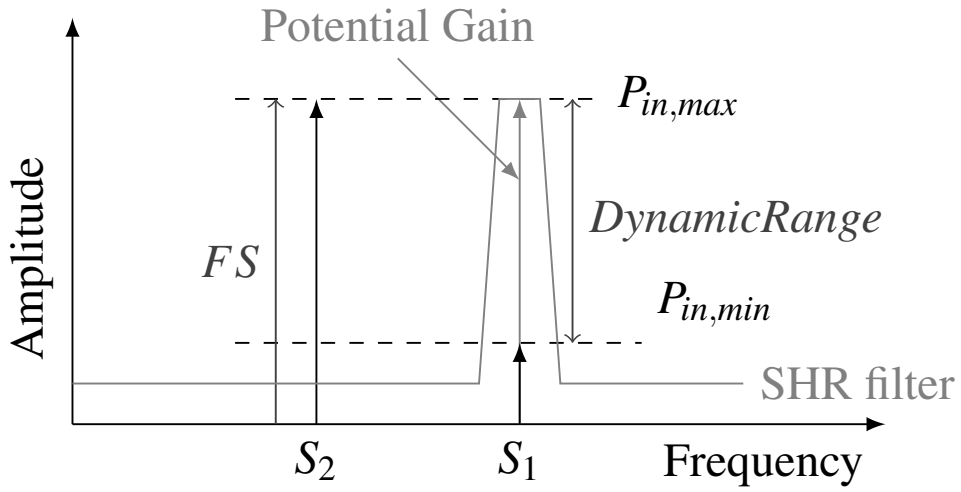


Figure 1.4: Dynamic range of two signals.

One of the main challenges for the FSR design, is the detection and extraction of a weak signal in the presence of (much) stronger signals. In figure 1.4 S1 represents the smallest detectable signal ($P_{in,min}$) and S2 represents the largest signal that we can receive $P_{in,max}$. The largest signal is defined by the maximum input voltage of the ADC. The smallest signal that can be detected defines the sensitivity of the ADC and is a result of the Least Significant Bit (LSB). The sensitivity is relative to the maximum input of the ADC and be-

cause it is limited by the quantization step, we can calculate it when we know the number of bits (N) with equation 1.1.

$$\mathrm{DR} = 20log_{10}\left(2^N\right) \tag{1.1}$$

If we want to receive S2 the ADC can sample the analog signal and it can be processed digitally. Signal S1 is represented by the LSB of the ADC, a single bit, which makes it very sensitive to errors caused by noise. On top of that the distortions caused by S2 can make it difficult to detect S1 as well.

In an SHR the analog filters are used to remove these strong interferers after reception. When S2 is filtered out, S1 can be amplified to match the full-scale of the ADC for the best reception. In the FSR there is no filtering before the ADC. This means that every signal, weak and strong, will appear at its input. Therefore it is important to determine how the FSR can deal with the reception of weak and strong signals at the same time.

On top of strong signals posing a challenge for the reception of small signals, Inter-Modulation Distortions (IMDs) caused by the harmonics can introduce problems as well. Due to nonlinearities of the LNA and the ADC, harmonics of these signals may be introduced and appear near or on top of the (small) signal of interest, causing distortions in the reception.

The bandwidth of each part in the receiver chain will put constraints and limitations on our receiver. The antenna, the LNA [21] and the ADC (mainly the Sample-and-Hold (S/H) circuit [22]) will each have a defined transfer function which influences the signal. Each of the individual parts introduces some noise which can be combined for the whole receiver chain in the total NF of the receiver. The characteristics of the components that we use to build the system, require attention when determining the feasibility of an FSR that fulfills our requirements.

The digital signal processing that needs to be performed is not trivial because billions of samples per second will be produced by the ADC. The FPGA possibilities may be used to its maximum and special design tricks could be necessary to implement the hardware that can handle this kind of high data-rates.

1.2.3 Costs

To define the cost of the FSR implementation we compare it again with the SHR. To receive the signals an antenna is required which matches the band. The SHR and the FSR generally require one antenna for each targeted frequency band. Only if multiple (far spaced) channels or bands are to be received, the FSR requires additional logic for combining the signals from multiple antennas. The SHR would "only" require a new antenna but excludes some signals for some of the receivers. For now we consider all bands of interest are received by the same antenna or includes a circuit for concurrent

reception with multiple antennas to prevent the exclusion of channels. Therefore we leave out the antenna from the equation for the cost function. Both systems require a demodulator, so we leave it out of the equation as well.
For an FSR system we require one amplifier, a good ADC and a DSP. When we want to make an SHR, we require 4 analog filters, two amplifiers, at least one mixer and an ADC.

For receiving additional channels with SHRs, we need a complete receiver per additional channel. This means that for each added channel, the cost per channel is increased by the cost of a single receiver. In the FSR implementation we require a second digital signal path, but no new ADC or other hardware is required, this gives it a much lower increase in cost.

We expect the traditional SHR to be a lot cheaper than the FSR implementation due to a lower graded ADC, which gives it an initial advantage. For each additional channel we expect that for the FSR the increase in hardware cost is lower for than for an SHR, as we show in figure 1.5. So there will be an break-even point after which it will be cheaper to use an FSR instead of an SHR with a certain amount of receiving channels.

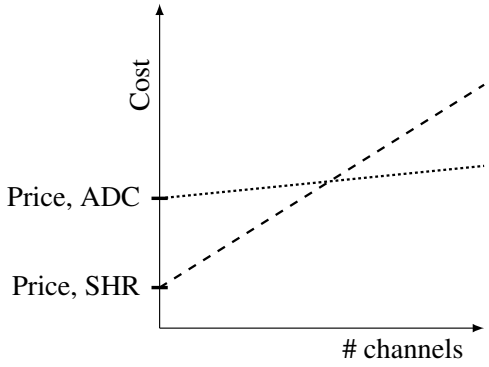


Figure 1.5: Cost functions of the FSR and the SHR.

1.2.4 Trade-offs

For ADCs there are typically 3 degrees of freedom: power, bandwidth and resolution, from which a number of possibilities and limitations can be derived. In this section we cover the trade-offs we can make between these parameters. We also present two alternative implementations that make an efficient exchange between these parameters using specific signal processing effects.

Figure of Merit

ADCs are generally defined by a Figure of Merit (FOM) by either $\mathrm{FOM}_{\mathrm{Walden}}$ (equation 1.3) or $\mathrm{FOM}_{\mathrm{Thermal}}$ (equation 1.4). This allows manufacturers and

scientists to compare ADCs. It also provides developers to make a rough estimation of the trade-off involved in ADC-design.

In [23] a classification is made to distinguish between different FOMs. They vary in the parameters they cover but almost all include the DR in some way. Usually the Signal-to-Noise and Distortion Ratio (SNDR) (equation 1.2) is used. This expresses the performance of the ADC in a relation between the maximum input ($P_{in,max}$) and the degradation caused by (quantization) noise (P_{Noise}) and nonlinearities of the system ($P_{Distortion}$).

$$\mathrm{SNDR} = \frac{P_{in,max}}{P_{Noise} + P_{Distortion}} \tag{1.2}$$

FOM_{Walden} states a one-to-one relation between resolution, expressed with the SNDR, and bandwidth: $+1$ bit $\propto 2 \times f_s$; this holds well for low-speed ADCs (≤ 100 [MHz]) up to gigahertz converters. $FOM_{Thermal}$ states that that bandwidth and resolution may be exchanged at a higher cost where each $+1$ bit $\propto 4 \times f_s$. This results in an exponential increase of the power if either is increased; this holds for high-speed ADCs (≥ 1 [GHz]).

$$FOM_{Walden} = G_{A1} = SNDR_{dB} + 20 \times log_{10}(\frac{f_s}{P}) \tag{1.3}$$

$$FOM_{Thermal} = G_{B2} = SNDR_{dB} + 10 \times log_{10}(\frac{f_s}{P}) \tag{1.4}$$

However, from [24] we observe that for the high-frequency (flash) ADCs the power dissipation increases tenfold for each additional bit. The FOMs presented in [23] appraise resolution accordingly to make a proper estimation of the trade-offs involved for an FSR implementation. Therefore we can not use these FOMs as they are a tool for ADC designers. Instead we have to consider the direct comparison between the DR and the f_s as a measure to define the performance of ADCs.

In figure 1.6 we present the relation between the resolution and the bandwidth of different ADCs presented at the International Solid-State Circuit Conference (ISSCC) and at the "VLSI and Embedded Systems Conference" [2]. For low speed ADCs we can see the 30 [dB] degradation of the SNDR per decade of bandwidth is correct according to FOM_{Walden}, but this does not hold for high speed ADCs where -70 [dB / decade] is a more suitable approximation.

Bandpass Sampling Receiver

Bandpass receivers, as shown in figure 1.7, use an analog filter before the sampling stage and mixing is performed digitally. They sacrifice (a portion of) bandwidth to reduce the required sampling rate, which directly results in a reduction in the power consumption of the ADC according to the equations of the FOMs. This works because of aliasing, which occurs when subsampling

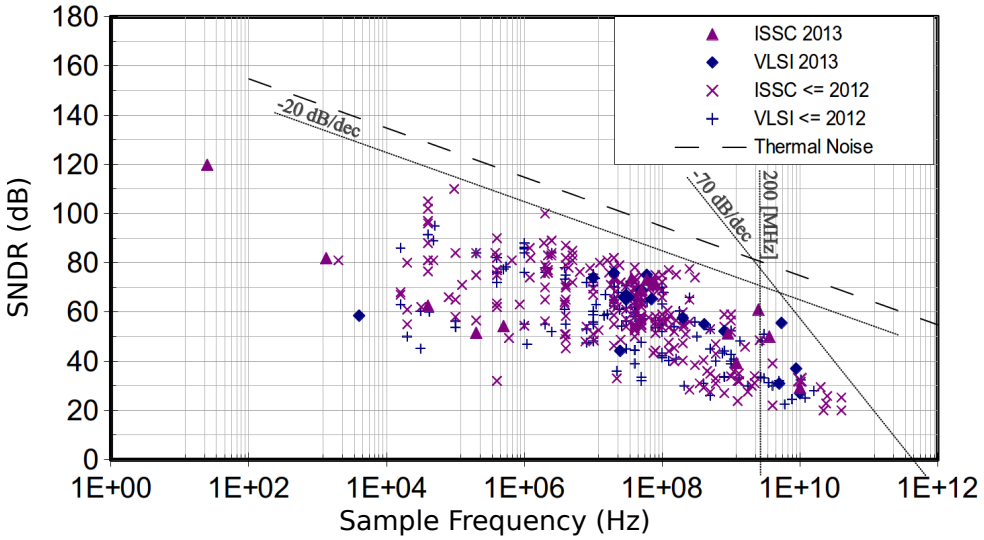


Figure 1.6: ADC survey of SNDR vs f_s from [2].

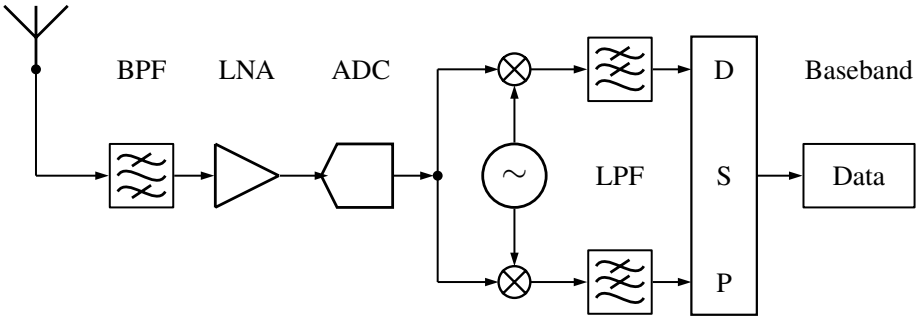


Figure 1.7: Bandpass receiver according to [3]

(f_s<f_c), where an image of the signal is projected in the baseband. Aliasing removes the need for a frequency shift so a mixer is not needed in this design.

Out Of Band (OOB)-interferers will be filtered out by the BPF so the requirements of the ADC DR are relaxed to only handle signal strength variations of the targeted signal. The challenge of this architecture is in the design of a good filter with a very sharp roll-off, which completely blocks any interferers while allowing the signal of interest to pass.

Direct conversion receiver

The Direct Conversion Receiver (DCR), shown in figure 1.8, remains similar to the original SHR architecture but leaves out the IF stage; the DCR performs the down-conversion to baseband in a single step.

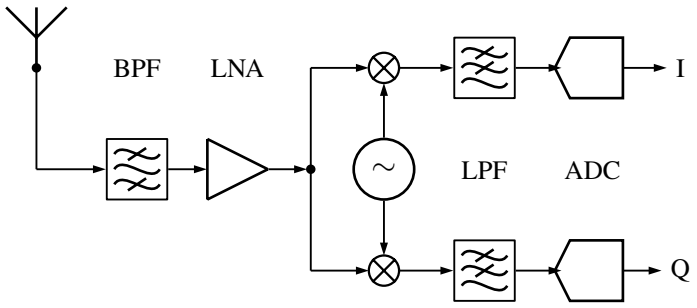


Figure 1.8: Direct conversion receiver according to [3]

The bandwidth of a direct conversion receiver is fixed due to its filters. The mixing stage may be implemented using a digital VCO which allows a wide reception range and good channel selection. Because of the analog filters the constraints on the ADC are relaxed. This architecture requires a very careful design to prevent a DC offset from the mixer to appear at the ADC.

1.3 Prototype Full-Spectrum Receiver

The components of our prototype are based around an ML605 FPGA development-board from Xilinx [25]. To its High Pin-Count (HPC)-connector, we connect an FMC125 daughterboard, developed by 4DSP, containing an 8-bit, 5 [GHz] ADC from e2v [26]. For the (wireless) reception of the signals, we have an LNA [21] available to amplify the RF signal before it is fed to the ADC. The signal from the ADC is processed in the FPGA and sent via an Ethernet cable to a Personal Computer (PC).

With this hardware we created an FSR which allows us to explore its possibilities and limitations. Due to the time available for this project we made trade-offs for the implementation of the digital signal processing.

1.4 Research questions

Following our wish to implement an FSR we formulated the following research question:

Can a Full-Spectrum Receiver, which meets our requirements for a practical receiver, be implemented using the available hardware for the prototype?

If not then:

1. what can we do to get as close as possible to our requirements?
2. what can we do to limit the design effort such that building the prototype fits in the scope of a graduation project?
3. given the current technology trends, when will it become possible to build a full spectrum receiver that meets our requirements?

1.5 Contributions

The main contribution of this thesis is a clear presentation on the requirements and the trade-offs involved in designing an FSR. The implementation with the available components, gives valuable insight about the reception characteristics and allows verification of the theory. After evaluating the results we could present the implications of the implemented receiver architecture. We present solutions to improve the receiver performance for future research. By studying the trends of ADCs we make a prediction when the implementation of an FSR that meets our requirements can become a reality.

1.6 Report outline

In this section we present the outline of our report. We begin by introducing the FSR architecture and present the minimum requirements for a practical receiver. To get an overview on the current state of the research on FSRs, we cover related work on receivers in chapter 2. We also present research from others on digital signal-processing because it is a very important aspect of the FSR. In chapter 3 we present the basis on which we defined our receiver specifications and give calculations to support our expectations. For the digital signal processing we present the considerations that will have the most impact on the performance of an FSR. In chapter 4 we present the FSR that we implemented with the available hardware and evaluate its performance. We draw conclusions of how our receiver can be improved and present our expectations for the future of FSRs in chapter 5.

Chapter 2

Related Work

This chapter presents the work of other researchers working on receiver technologies that are used by or closely related to Full-Spectrum Receivers (FSRs). First we cover the research on several conventional receiver architectures. The second part of this chapter covers work on digital signal processing techniques.

2.1 Receiver Architectures

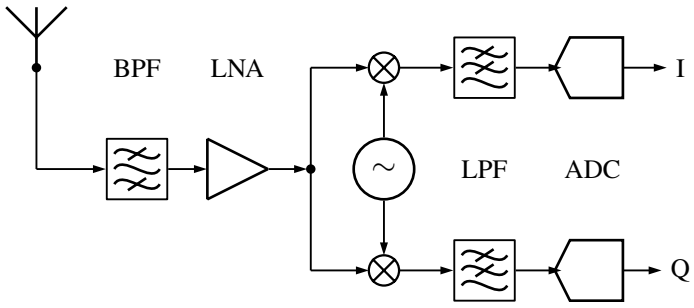


Figure 2.1: Direct Conversion Receiver topology from [3].

In this section we present two other receiver architectures which use analog circuits to overcome the issues that the FSR suffers from and work that is performed on FSRs. Other receivers sacrifice bandwidth to enable the concurrent capturing of their signals. A comparison of the techniques is given in [1].

2.1.1 Direct Conversion Receivers

The Direct Conversion Receiver (DCR)[1], figure 2.1, originates from the homodyne receiver topology [27]. The Radio Frequency (RF) signal is first filtered to remove Out Of Band (OOB) and especially High-Frequency (HF) signals that could alias back to baseband. Then it is mixed to baseband using an

[1]Also referred to as "zero-IF"or "direct-down conversion"receiver

oscillator running at Carrier Frequency (f_c), so the Analog-to-Digital Converter (ADC) may sample at a frequency much lower than f_c (but at least at twice the bandwidth). This receiver topology has the serious drawback of introducing a Direct Current (DC)-voltage drift, caused by mixing the input with the local oscillator. This can be a major issue for some radio standards as presented in [28]. Nevertheless DCRs are used more often in receivers, for example in mobile handsets, due to their low part-count and complexity compared to the Super-Heterodyne Receiver (SHR).

2.1.2 Bandpass Receivers

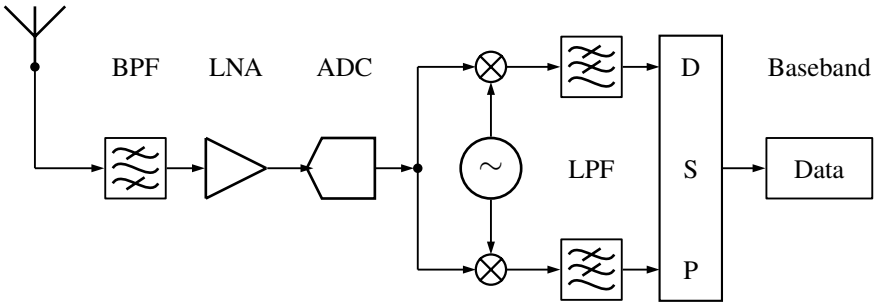


Figure 2.2: Bandpass receiver topology from [3].

The Band-Pass Receiver (BPR)[2] (figure 2.2) makes a very efficient trade-off between bandwidth and sample frequency (f_s)[3] by using aliasing (caused by subsampling). The BPR requires a filter with a very sharp frequency roll-off (high quality factor) to allow this phenomena to occur without destroying the received signal with aliases and OOB interferers. Aliasing is already described in the early days of digital signal processing [29]. Due to the technical difficulties of implementing the required filters it took a long time before practical subsampling (bandpass sampling) became a possibility [30] and was applied in systems [19].

Recent studies mostly focus on the reception of multiple bands using a single ADC [31] [32] [18]. The drawback of this application is that it requires distinct RF-filters for each band combined with careful alias-frequency-planning. This means the minimum sample frequency [33] is not always achieved in such systems and overdimensioning the component capabilities is required.

2.1.3 Full-Spectrum Receivers

We are looking to employ an FSR for wireless communications. Chips and products that use an FSR-architecture already exist for wired systems. Com-

[2]Also referred to as subsampling receivers.

[3]A lower f_s generally results in a lower power consumption.

panies have successfully implemented FSRs in their products, replacing the expensive analog tuners [14] [13]. We found claims that Digital Video Broadcasting (DVB)-satellite FSRs already exist, but little information could be found and demonstrations are planned for later this year [34].

2.2 Multirate Digital Signal Processing

In this section we discuss the digital processing part that provides the demodulator with the baseband signal that it requires. Typically a Digital Down Converter (DDC) [10] is used to translate the digital signal from RF to baseband. It consists of two parts, a mixer and a low pass decimating filter as depicted in figure 2.3. These parts do not enforce a specific implementation, so there are multiple variants available for each.

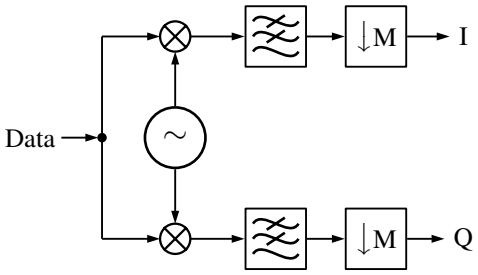


Figure 2.3: Architectural overview of the DDC.

2.2.1 Digital Down-Converter Mixing Stage

For the FSR the DDC is a critical component because it has to mix a signal from RF, which is generally in the gigahertz domain, down to baseband, around DC. While the filter may perform down-sampling, the mixer may not, because it can destroy the information contained in the signal, so mixing will have to be performed at the same rate as the f_s.

With a Numerically Controlled Oscillator (NCO) a Direct Digital Synthesizer (DDS) can be made to create a sine and cosine signal. The mixing is performed at such high speeds that usually Application Specific Integrated Circuits (ASICs)s are designed for this task and the mixing performed by an external chip [35] [36] instead of inside the Field-Programmable Gate Array (FPGA). Implementations do exists for FPGAs from Xilinx [37] using a DDS but the maximum speed of the Xilinx-core is limited (≤ 450 [MHz]). Therefore researchers have come up with custom DDS-implementations for FPGAs [38].

It is also possible to store all the values of a sine (and cosine) in a Look-Up Table (LUT) and multiply them with the signal. Multiplication techniques have been developed and optimized for the FPGA over the years [39]. The

drawback of this multiplier-less implementation is the not parallelizable (sequential) implementation of one step in the algorithm. This creates a bottleneck limited by the speed of the platform, rendering implementations for gigahertz sampling rates infeasible[4].

In [5] 128 parallel multiplicators are implemented to handle GHz sampling rates and mix the signal to baseband using a LUT. With a high f_s, to mix low frequency signals to baseband, we require an big LUT. Therefore the main drawback of this implementation is the need for a large memory to cover all frequencies.

Another implementation to mix the signal to baseband is the COrdinate Rotation for DIgital Computers (CORDIC) algorithm which has been extensively developed [40]. The size of the core (in an FPGA) is similar to that of a multiplier [41]. The implementation is multiplier-less and calculates the rotation of the I and Q signals to perform the frequency translation (to baseband). The CORDIC core performs complex signal processing in a single core, compared to the four multiplicators which are required when processing a complex signal with a LUT implementation. This algorithm can be performed in parallel and does not require as large a LUT as an NCO with multiplier implementation requires, making CORDIC a well suited implementation for use on FPGAs.

2.2.2 Digital Filter Design

The FSR requires all the filtering to be performed in the digital domain, which means that filters are relatively easy to design, but they can become too large for the FPGAs on which they should be implemented. The traditional Cascaded Integrator-Comb (CIC)-filter can not be implemented in parallel and thus its throughput is limited by the platform speed. This renders the use of this class of filters useless when implementing an FSR. A recent implementation [5] uses multiple (cascaded) Finite Impulse Response (FIR)-filtering stages, which can be put to work in parallel. The clever use of the platform specifications to perform half-band filtering with basic-components allows them to implement three DDCs in a single Virtex-6 FPGA[5].

[4] If the sample rate is reduced to a manageable speed with other techniques, this implementation may still prove very useful.

[5] For our prototype we will also use a Virtex-6 FPGA.

Chapter 3

Design Considerations

In this chapter we cover the implications and requirements for a Full-Spectrum Receiver (FSR) system. We start at the front of the receiver chain by defining the signal we want to receive with the antenna. After that we cover the effects of the parts up to the Analog-to-Digital Converter (ADC). Following that we cover the digital signal processing which is required to present the signal to the back-end (demodulator). With the mentioned considerations for the FSR we can draw conclusions on the applicability of such a system.

3.1 Reception Characteristics

In this section we present the calculations and properties of the signal and hardware up to and including the ADC. With these calculations we can define the expected performance of our hardware in a receiver. For reference purposes we express the requirements that we found for practical receiver parameters again from chapter 1.

3.1.1 Signal Specifications

Our goal is to design an FM-receiver for mono-audio radio signals. In figure 3.1 we see that the audio signal is a 15 [kHz] band. This signal will be frequency modulated with the maximum deviation $\Delta f = 75$ [kHz] resulting in total bandwidth (B) of the modulated signal $B_{FM} = 150$ [kHz]. This signal should have Signal-to-Noise Ratio (SNR) ≥ 10 [dB], otherwise demodulation will not be possible [42]. The use of an initial amplification is considered to have a positive effect on the reception of Radio Frequency (RF) signals, therefore we will make use of an Low-Noise Amplifier (LNA).

We defined the following desired characteristics for a practical receiver:

- Maximum (blocker) signal strength: $P_{in,max} = -10$ [dBm]
- Minimum signal strength (sensitivity): $P_{in,min} = -90$ [dBm]
- Dynamic Range (DR): $P_{in,max} - P_{in,min} = 80$ [dB]
- Gain: $Gain_{LNA} = 20$ [dB]

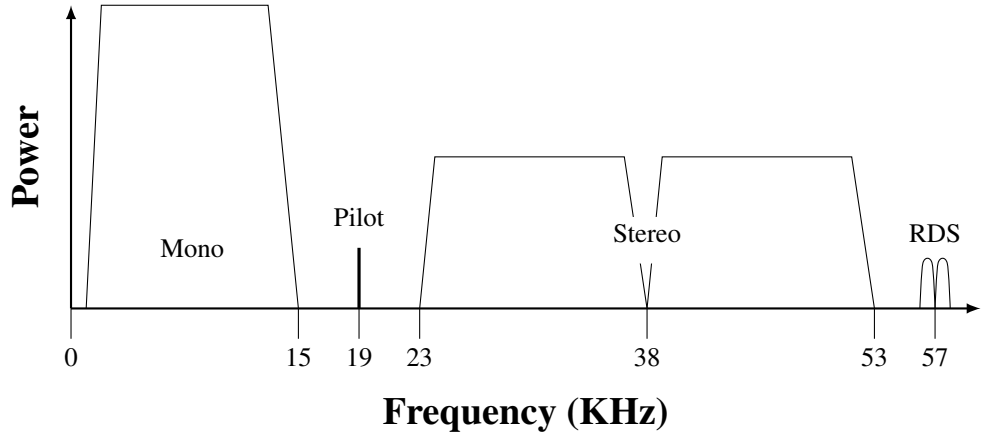


Figure 3.1: FM-baseband spectrum, from [4].

3.1.2 Maximum Received Power

The signals that will be received are bounded by the strongest signal in the spectrum, these signals are therefore called *blockers*. Following the research presented in [16] we assume they have a maximum power of -10 [dBm]. Equation 3.1 presents the relation between the voltage range of the ADC ($V_{p\text{-}p}$), the termination resistor (R) expressed in Ω and the maximum input power ($P_{in,max}$) expressed in dBm.

$$P_{\text{in,max}} = 10log_{10}\left(\left(\frac{V_{\text{p-p}}}{2\sqrt{2}}\right)^2 \times \frac{1000}{R}\right) \text{ [dBm]} \tag{3.1}$$

The LNA provides an amplification of ~ 20 [dB], therefore the ADC will receive $P_{in,max,ADC} = -10 + 20 = 10$ [dBm]; with a 50 [Ω] termination resistor and rewriting equation 3.1 this results in a required $V_{in,max,ADC} = 2$ [$V_{p\text{-}p}$]. For a DR of 80 [dB] we require at least $log_2(10^{\frac{80}{20}}) = 13.3$ bits [43].

$$\text{SNR} = 6.02 \times n + 1.76 \text{ [dB]} \tag{3.2}$$

Our ADC has 8 bits and following equation 3.2 from [44], we find the best $SNR_{ADC} = 49.9$ [dB] [43]. The input voltage range of the ADC $V_{ADC,p-p} = 500$ [$mV_{p\text{-}p}$]. With equation 3.1 we find the maximum input power of the ADC $P_{in,max,ADC} = -2$ [dBm].

Therefore the power delivered to the LNA may be at most $P_{in,max,LNA} = -22$ [dBm]. The LNA begins to exhibit more than 1 [dB] nonlinear behavior when the input power exceeds 3 [dBm], the so called ˝1dB-compression point˝. As long as we do not exceed the input power domain of the ADC then we can expect the LNA to behave linearly.

3.1.3 Minimum Received Power

The smallest signal that we can detect is defined by the noise floor, which is a combination of the thermal noise (N_T), quantization noise and distortions. Thermal noise is caused by the motion of electrons in an electric system and causes -174 [dBm] per hertz at an antenna temperature of 293 [K]. Quantization noise is caused by the number of bits used for the representation of a signal. Other distortions are caused by nonlinearities in the LNA and ADC and are calculated using the Inter-Modulation Distortion (IMD)-characteristics of the systems.

3.1.4 Thermal Noise

The FM signal band is 200 [kHz][1] wide. We use equation 3.3 where we fill in Boltzmann's constant ($\mathrm{k_B} = 1.3806488e-23$ [J/K] [41]), the antenna noise temperature (T= 293 [K]) and the bandwidth (B) to find the thermal noise floor in this band $N_T = -101$ [dBm].

$$N_T = 10log_{10}(\mathrm{k_B TB} \times 1000) \text{ [dBm]} \tag{3.3}$$

The thermal noise is uniformly spread white noise in the received spectrum and therefore affects the input SNR. The SNR of a received signal has to be high enough for a demodulator to process the signal. The thermal noise floor puts a fundamental limit on the minimal noise power, this enforces a minimum power for the received signal. In our FSR we will not be limited by the thermal noise because the noise floor for 2 [GHz][2] lies at $10log_{10}(\mathrm{k_B} \times 293 \times 2e9) = -80.82$ [dBm] which is below the sensitivity of our ADC.

To calculate the minimum required signal power we need to define the receiver Noise Figure (NF). For cascaded stages we can calculate the total NF with Friis' formula for noise, shown in equation 3.4. The total NF will never get lower than the NF of the first stage of the receiver, so a low NF is desired. This relation also indicates that the gain of the first component can be used to "suppress" the NF of components further in the chain. Thus a high gain in the first step is beneficial for the reception because gains later in the chain are less effective.

$$\mathrm{F_{Total}} = \mathrm{F_1} + \frac{\mathrm{F_2} - 1}{\mathrm{G_1}} + \frac{\mathrm{F_3} - 1}{\mathrm{G_1 G_2}} + ... \tag{3.4}$$

$$\mathrm{NF} = \mathrm{P_{FS}} + \mathrm{N_{T,1Hz}} - \mathrm{SNR} - 10log_{10}(\mathrm{B}) \tag{3.5}$$

All these values must be used in their linear representation (we generally use the logarithmic representations in dB). The NF and the gain of the LNA

[1] Because the actual band for our signal is a bit over 150 [kHz], we can take 200 for the ease of calculations because it is a worse-case.

[2] This is the bandwidth of the Sample-and-Hold (S/H)-circuit of the ADC.

are given by the datasheet [21]. For the NF of the ADC we use the calculations from [45] in equation 3.5 where $P_{FS} = 2$ [dBm]; $N_{T,1Hz} = -174$ [dBm]; SNR $= 45$ [dB]; $B = 2$ [GHz]. We calculate the linear representations of the values we already know and obtain the following values:

- $NF_{LNA} = 3$ [dB] $\rightarrow F_1 \approx 2$
- $Gain_{LNA} \approx 23.5$ [dB] $\rightarrow G_1 \approx 100\times$.
- $NF_{ADC} = 33$ [dB] $\rightarrow F_2 \approx 2000$

With these values and equation 3.4 we find for our receiver that $NF_{Total} \approx 10$ [dB].

3.1.5 Quantization Noise

Quantization is an effect that occurs in the conversion from a continuous valued signal to a discrete-valued signal. This occurs in digital signals because they are represented by a finite number of bits. This transformation introduces some loss of the signal's information, which can be seen in figure 3.2. We used 3 bits for the representation giving us 8 quantization levels to show the effect.

The maximum introduced error by an ADC is $\pm\frac{q}{2} \times V_{max}$, where q represents the smallest representable value, $q = \frac{1}{2^N}$ and N is the number of bits used. The Root Mean Square (RMS) value of this error is $V_{max,error} \times \frac{q}{\sqrt{12}}$. The resulting SNR can be calculated using equation 3.6 and is usually generalized to equation 3.7 [44].

$$SNR_{quantization} = 20 \times log_{20}(\frac{V_{in,RMS}}{V_{noise,RMS}}) \text{ [dB]} = 10 \times log_{10}(\frac{P_{in}}{P_{noise}}) \text{ [dB]} \quad (3.6)$$

$$SNR_{quantization} = 6.02 \times N + 1.76[\text{dB}] \quad (3.7)$$

Our ADC uses 8 bits for the digital representation of the sampled signal and has a maximum input voltage swing (full-scale) of $V_{FS} = 500$ [$mV_{p\text{-}p}$] resulting in a quantization step between two quantization levels of $\frac{.5}{2^8}[mV] \approx 1.95$ [mV]. The SNR of our ADC is equal to $SNR_{quantization} \approx 50$ [dB] [46].

3.1.6 Inter-modulation Distortions

The IMD can be significant in receiver systems, causing distortions much stronger than the thermal noise floor. IMD is caused by the nonlinearities in the used components [47]. Because it is impossible to have completely linear parts, we have to define what the impact of the IMD will be on our receiver.

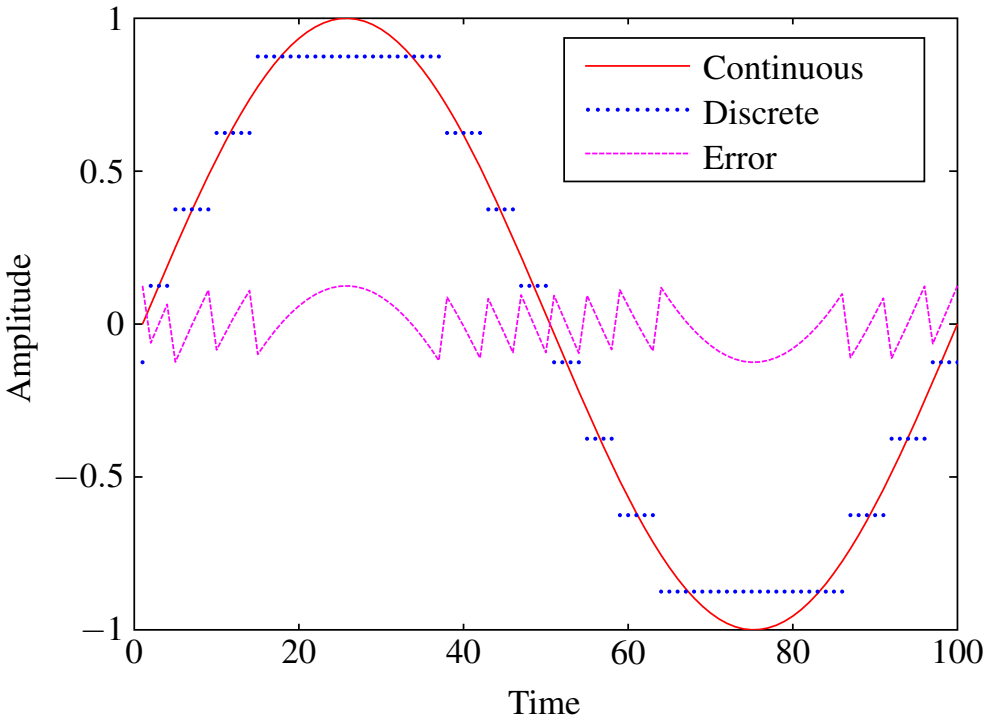


Figure 3.2: Quantization noise caused by representing a continuous valued signal with a 3 bit valued signal.

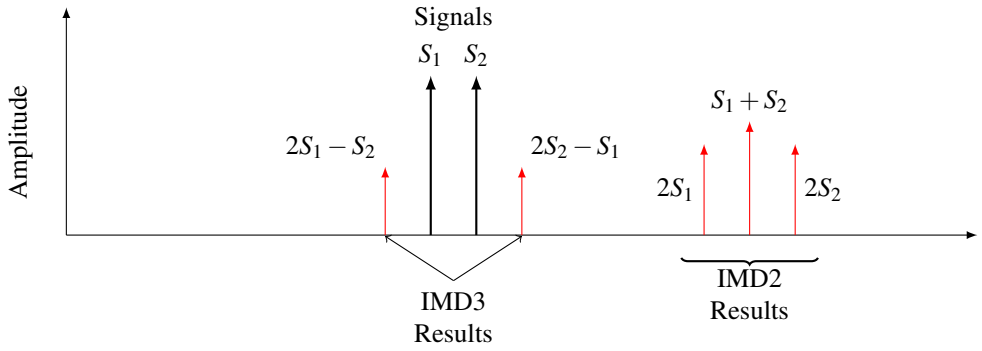


Figure 3.3: Inter-modulation distortion of two sine waves.

The IMDs are proportional to the carrier signal strength and therefore usually expressed in [dBc], this means they will become smaller with smaller signals.

In figure 3.3 we show the frequency spectrum containing the second and third order IMDs (IMD2 and IMD3) of two sine waves. The IMD2 signals are located far from the original signals and with proper filtering should pose no

problem. The IMD3 distortions however, appear close to the original signals, as can be seen and this can cause distortion of the wanted signal[3]. Because they appear so close to the targeted signal it is hard to remove them with filtering which makes a low IMD very important for receivers.

In most conventional receiver techniques the IMD2 (and higher order even IMDs) appear far away from the original signal and are removed by analog filtering. Due to the large bandwidth of the FSR, even IMDs may appear close to other signals [48] and distort them. Meaning that a desired signal can be distorted by the IMD caused by other signals.

3.1.7 Spurious Free Dynamic Range

One way to express the IMD caused by a receiver is to express its Spurious-Free Dynamic-Range (SFDR). This defines a relation between the power of these IMD spurs relative to the signal power (f_c) or the full-scale of the ADCs.

The SFDR of our ADC is given in the datasheet $\text{SFDR}_{\text{ADC}} = 48$ [dBc]. Mixers and amplifiers have a generalized model to determine the IMDs with so called intercept Point (IP) . The second and third order IP (IP_2 and IP_3) are used to calculate the performance. The input of the ADC will clip with too strong signals [49] and this makes the ADC too nonlinear to make use of this model. Therefore we have to review the effects of the separate components of our receiver chain.

With an expected maximum signal strengths of -2 [dBm] the spurs from IMD in the ADC are expected to have a maximum strength of $\text{P}_{\text{Spurs,max}} = \text{P}_{\text{in,max,ADC}} - \text{SFDR}_{\text{ADC}} = -50$ [dBm] 3.4. Note that if no signals with -2 [dBm] are present at the input of the ADC, the spurs will be weaker but also the full-scale is used inefficiently. For Frequency Modulation (FM) we require $\text{SNR}_{\text{FM}} \geq 10$ [dB] which means that the weakest detectable FM signal will have to be -40 [dBm] at the input of the ADC, otherwise it may be distorted by IMDs.

With the $\text{IP}_{3,\text{LNA}} = 14$ [dBm] we can calculate the IMD3 caused by the LNA. Using equation 3.8 from [48], where the (thermal) noise floor $N_0 = -81$ [dBm] with a bandwidth B = 2 [GHz], we find $\text{SFDR}_{\text{LNA}} \approx 64$ [dB]. This means that the input power to the LNA will be limited by the ADC input range. Therefore we expect that the LNA to have little negative influence on the total SFDR of our system.

$$\text{SFDR}_{\text{dBm}} = 2/3(\text{IP}_3 - N_0) \text{ [dBm]} \tag{3.8}$$

[3]Higher order uneven IMDss (5, 7, etc.) can also end up close to the original signal but are much weaker.

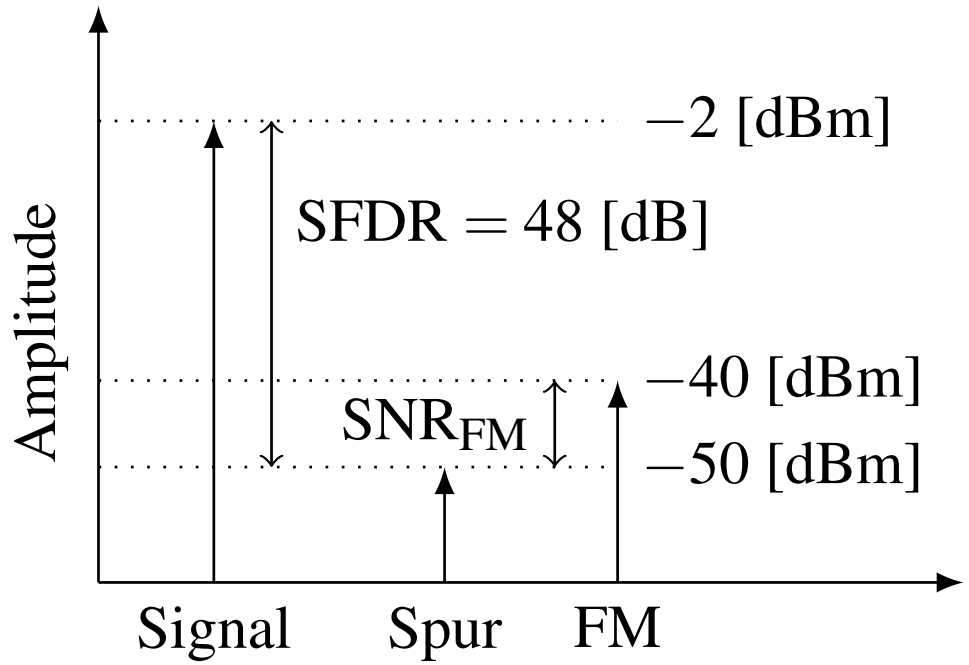


Figure 3.4: Maximum signal, spur and minimum FM signal strengths.

3.1.8 Summary

The characteristics of the ADC impose the dominant limitations on the received signal. $P_{max,ADC}$ pushes the maximum received power on the LNA to ≤ -22 [dBm]. On the lower side the thermal noise floor enforces a minimum power of -81 [dBm], but before that, the ADC reaches the soft-limiting bound $P_{Spurs,max} = -50$ [dBm] caused by the SFDR and 2 [dB] lower the hard-limited bound caused by the quantization. Add to that a gain of 20 [dB] from the LNA, resulting in signal strengths that can be received by the LNA between -70 [dBm] and -22 [dBm] before nonlinearities occur and the signals are destroyed. For the demodulation of FM signals we require an SNR of at least 10 [dB], so the minimum power of the signal we receive should be -60 [dBm].

3.2 Digital Signal Processing

In this section we discuss the implementation of the digital signal processing in an Field-Programmable Gate Array (FPGA) and the requirements for an FSR. In chapter 2 we described several optional implementations. For an FSR we can define the required performance of the digital processing, find trade-offs for the implementation and express the implication of these choices.

Following the traditional Digital Down Converter (DDC) architecture, shown in figure 2.3, we have to perform a frequency translation to baseband and filtering, not necessarily in that order. We have the option to perform the conversion to baseband using a mixer or by downsampling. For the filtering we can use low pass, high pass and bandpass filters; which may be cascaded to create a distinct frequency plan. Different filter implementations may be considered but we can already rule out the use of Cascaded Integrator-Comb (CIC)-filters [50] because they can not match the speed for the desired sampling rates. Infinite Impulse Response (IIR)-filters are also unusable in the FSR because of their internal feedback mechanism which limits the implementation to the platform speed. The platform speed of our FPGA is ≤ 600 [MHz] while sample frequency (f_s) will be at least twice as high.

3.2.1 Frequency Translation Using Decimation

A high-speed RF-Finite Impulse Response (FIR) filter could be implemented to use downsampling (decimation) as a way to mix the signals to baseband with the aliasing effect. With Filter Design and Analysis-Tool (FDA-Tool) from Matlab we designed a suitable bandpass (equiripple) filter following [51]. We use the parameters below to obtain a passband of 200 [kHz], just enough for an FM signal and an Out Of Band (OOB) attenuation of 13 [dB].

- Fs: 1.25 [GHz]
- Fstop1: 100.05 [MHz]
- Fpass1: 100.1 [MHz]
- Fpass2: 100.3 [MHz]
- Fstop2: 100.05 [MHz]
- Astop1: 13 [dB]
- Apass: 1 [dB]
- Astop2: 13 [dB]

The resulting filter is shown in 3.5, which perfectly displays the size of the passband compared to the total bandwidth. This filter performs the filtering and frequency shift to baseband in a single step but requires 16663 multiplications per sample. Our FPGA contains 768 DSP-slices to multiplications, which means this filter design is infeasible for our sample rate.

Therefore we have to use an alternate implementation to perform the filtering. We suggest to use cascaded filtering stages because that way we can utilize the advantages of each sub-filter (power, space, speed) without losing in total performance [52].

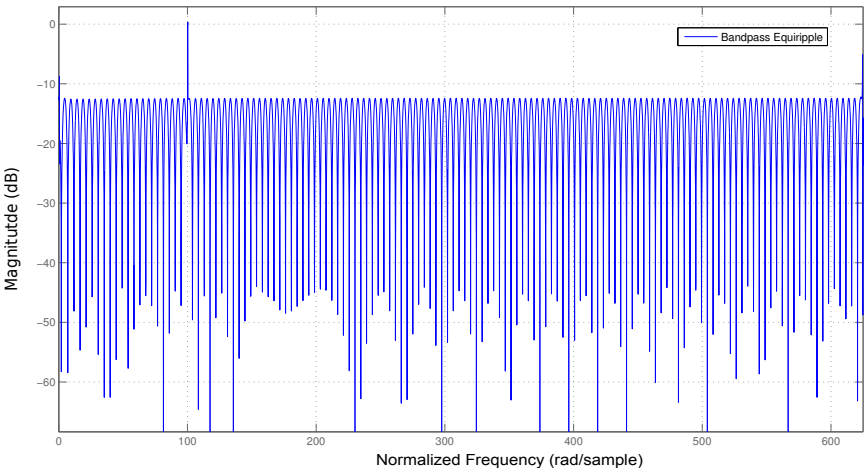


Figure 3.5: High order filter providing a 200 [kHz] bandpass filter at f_s = 1.25 [GHz] with 13 [dB] OOB attenuation.

Using cascaded filters allows the use of aliasing in combination with filtering, but signals that are not mixed to baseband require bandpass filters. This means that a low- and a high pass filter will be required, effectively doubling the amount of filter-taps (multiplications) required. Therefore it is undesirable to only use for the down-conversion of a signal.

3.2.2 Decimating FIR-filter Implementation

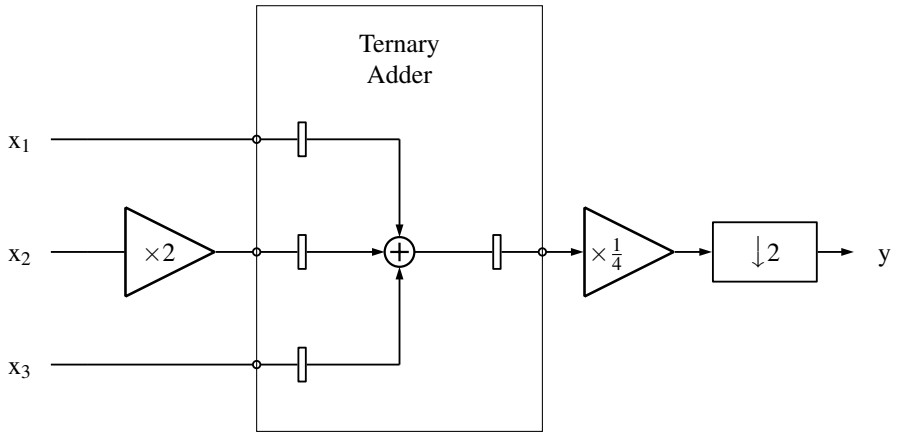


Figure 3.6: Extended version of half-band filter from [5].

In figure 3.6 we show the architecture for a half-band filter with 3 taps which is presented in [5] for implementation in a Virtex 6 FPGA. The inputs

of this filter are the raw values from the ADC, 8-bit values for each sample. x_2 must be multiplied by 2, to realize this, we shift in a '0' on the right of the vector increasing the vector to 9 bits.

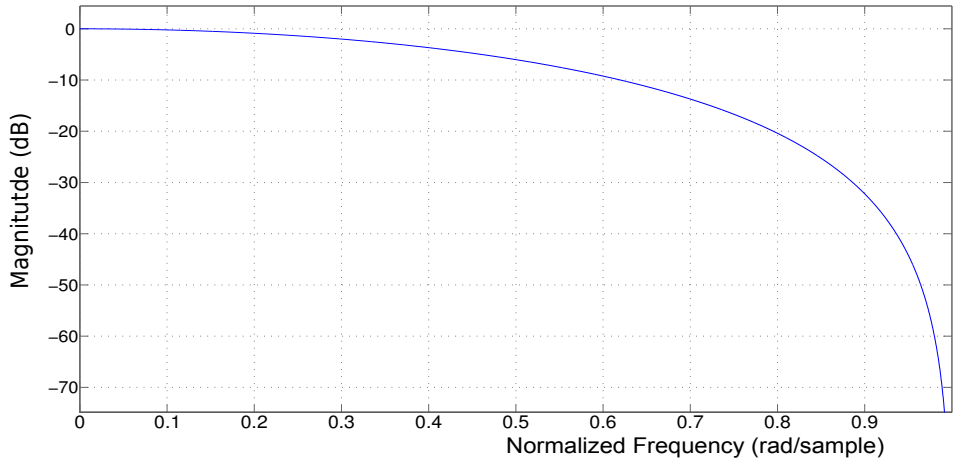


Figure 3.7: 3-taps half-band filter transfer function created using FDA-Tool.

The inputs are added together with a ternary adder, registers may be used to pipeline the process but care must be taken that the number of registers remains the same for all parallel placed adders. The result will grow by 3 bits to N+3 because of the shift and addition. It must be divided by 4 to prevent unwanted gain, this can be performed in a fixed-point representation by shifting the point two places to the left. With the FDA-Tool we find the transfer function of this filter presented figure 3.7.

This filter is minimal in size, capable of running at a high speed, parallelizable and provides a -3 [dB] cut-off frequency at $\frac{f_s}{4}$. All these properties make it an ideal first step for a cascaded filter structure, but only if the targeted signal is below the cut-off frequency.

3.2.3 Frequency Translation With Mixing

Applying a frequency down conversion of a signal is usually performed by multiplying the input signal with a complex Carrier Frequency (f_c) to translate it to baseband. In [5] a Look-Up Table (LUT) is used to store this complex sine wave for the carrier frequency. While actually only 1/4th of a sine wave is required to create a full sine-wave, the logic will have to work at the f_s of the ADC. The lowest f_c will define the size of the LUT. To translate a signal with f_c= 200 [MHz] and f_s= 5 [GHz] to baseband we require of a full sine wave containing $\frac{5e9}{200e6} = 25$ samples. With a platform clock frequency of 600 [MHz], at least $\lceil \frac{25}{600e6/200e6} \rceil = 9$ complex multipliers must be placed in

parallel to compute the multiplications for this implementation. In practice this amount will probably be higher because the maximum platform rate of 600 [MHz] may not be achieved.

The required memory has to be implemented for each multiplier, because each multiplier needs to access a different value of the sine-wave at the same time. The memory must contain $5e9-1$ samples to be able to mix frequencies with a carrier frequency between Direct Current (DC) and $\frac{f_s}{2}$ to baseband. If we can make number of values equal to (a multiple of) the number of multipliers (N); then we only need to store 1 in N values for each multiplier.

If we consider a more practical implementation the constraints on the processing may be further reduced. With a minimal f_c of 400 [kHz] we only require one in $\frac{5\ [\text{GHz}]}{10\times 400\ [\text{kHz}]} = 1250$ samples. To store the sine waves as 14-bit values ($\text{SNR}_{Quantization} \approx 84.2$ [dB]) we require $2500 \times 14 = 17500$ [kbit] of storage. This fits in two 18 [kbit] Xilinx block-Random-Access Memorys (RAMs) [53] and with 9 multipliers we require 18 of these to store the values for the I and Q paths.

In similar research we found that using a COrdinate Rotation for DIgital Computers (CORDIC)-algorithm [54] can provide a viable solution in FPGAs. In [55] we find that the maximum achieved rate for a CORDIC implementation on our FPGA to be ~ 340 [MHz]. Considering a sampling frequency of 5 [GHz] we require 16 cores in parallel to perform the down-mixing of a signal to baseband. The CORDIC implementation works with an accuracy of up to 16 bits.

Both implementations scale linearly with the data-rate to a minimum of 1 core if digital signal processing is applied before mixing. The size of a complex multiplier is about the same size as a CORDIC-core. The memory required for the LUT is higher for the multiplier implementation but considering only practical applications this may not be a problem.

3.2.4 Summary

There are many techniques and tools available to provide the required digital filtering and frequency translation for the FSR. Multistage filtering is necessary to cope with the high data rates because performing the required filtering in a single-stage requires too much resources. With CORDIC-cores we can translate any arbitrary signal from our input to baseband while a LUT implementation requires optimization steps that limit the flexibility. When the mixing is done before the down-sampling, following the classic DDC-architecture, we can use Low-Pass Filters (LPFs) instead of Band-Pass Filters (BPFs).

By placing decimating filters before the mixing stage we can apply down-sampling through decimation. This relaxes the constraints of the mixing stage but limiting the input frequencies. The amount of cores for the mixing stage may be reduced because of the lower sampling rate. When performing this pre-conditioning of the signal before the mixing stage, an interesting trade-off

can be made between the required resources of the FPGA and the capabilities.

3.3 Conclusions

By looking at the reception characteristics of our system and comparing them to the requirements, we identified points that require special attention when designing an FSR. The DR of the ADC is very important because it defines the signal strengths that may be received; for the desired DR of 80 [dB], we require 13.3 [bits]. The minimum received signal strength is restricted by the minimal SNR for demodulation and the bandwidth of the signal due to the thermal noise, but only when proper filtering is applied. The maximum received signal is defined by the voltage swing of the ADC and the amplification of the LNA. Digital signal processing for FSRs is a demanding job because of the high data rates; while the FPGA provides numerous solutions for parallel implementations to solve the speed problem, the design of an optimized architecture for digital signal processing architecture is still a non-trivial task. We regarded the required effort for implementing a proper filter in the FPGA and decided this would probably consume too much time for this project. Therefore we will present the implications of this trade-off in chapter 4 and focus on the other properties of the FSR.

Chapter 4

Full-Spectrum Receiver Design

In this chapter we present the receiver we implemented to test and verify our analytical results in practice. The main goal is to create a wireless FM-radio receiver for frequencies up to 2.5 [GHz] by connecting the provided Analog-to-Digital Converter (ADC) and Field-Programmable Gate Array (FPGA)-board to a Personal Computer (PC). First we give an overview of the used hardware and the test equipment. This provides insight of the system shows the limitations and bottlenecks. Then we discuss the choices we made and give an overview of the implications. The realization and verification steps of the system are presented to draw conclusions about our implementation compared to the Full-Spectrum Receiver (FSR)-design.

4.1 Receiver Hardware

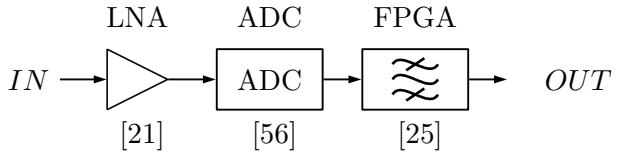


Figure 4.1: Designed system hardware overview.

In this section we give an overview of the hardware that we used to implement and test our receiver. In figure 4.1 we show the receiver architecture that we use for our system. We also present the transmitter devices that we used to create a signal for our receiver.

4.1.1 LNA: ZFL-1000LN+

According to the Friis equation, a high gain early in the receiver chain greatly benefits the reception [45]. We place the LNA as the first block of the receiver chain to make the most use of it.

The LNA has an in- and output using SMA-connectors impedance matched at 50 [Ω] to both the antenna and the ADC. The maximum input power is 5 [dBm] which is well within the maximum received power strength we defined: −10 [dBm].

The directivity of > 18 [dB], stated in the datasheet, is an indication of its linearity [57]. This is considered good enough to not have to worry about it in our system. The supported bandwidth is defined from 100 [kHz] to 1 [GHz]. The provided gain is typically 20 ± 0.5 [dB] at $V_{supply} = 12$ [V] with a Noise Figure (NF) of ~ 3 [dB].

4.1.2 ADC-board: FMC125

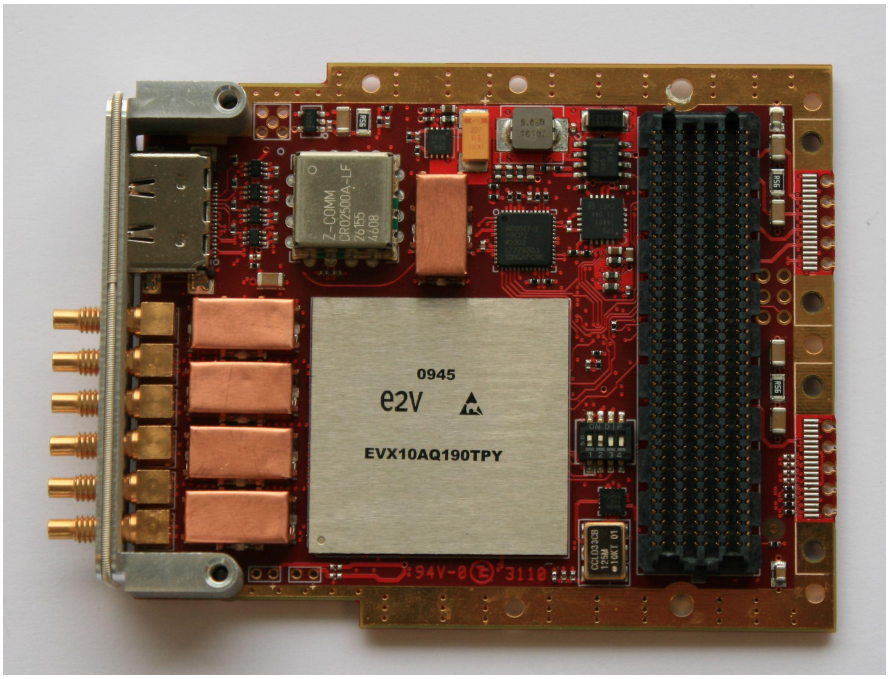


Figure 4.2: FPGA Mezzanine Card, FMC125 [6].

The FMC125 [56] daughter board that is used in this research contains an EV8AQ160 ADC [26]; a picture can be seen in figure 4.2. Figure 4.3 provides a schematic overview of all the inputs (left side) and outputs + control (right side), as well as a rough sketch of the internals.

The board has four Sub-Miniature version C (SMC)-connectors for input signals, terminated with a 50 [Ω] resistance. Each of these channels contains an ADC that can sample at frequencies between 200 [MHz] and 1.25 [GHz]. When feeding the same signal to all four inputs, a combined sampling rate of $f_s = 5$ [GHz] may be achieved. The input bandwidth per channel is defined by the track-and-hold (T/H) circuit which has a programmable bandwidth

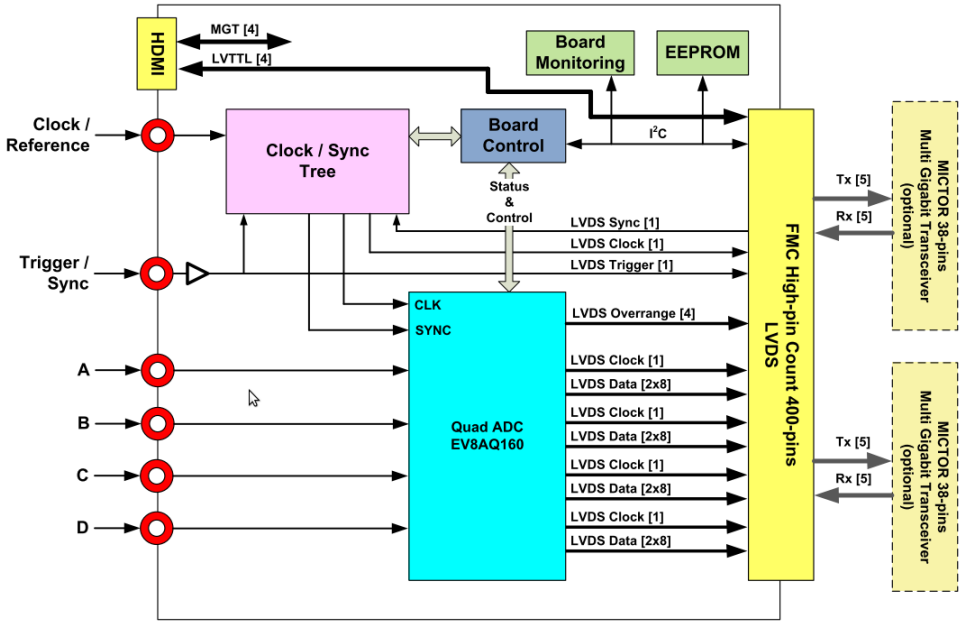


Figure 4.3: FMC125 block diagram from [6].

of 500, 600, 1500 or 2000 [MHz]. The maximum input voltage for each of these inputs is 500 [$mV_{p\text{-}p}$] with an input resistance of $R_{in} = 50$ [Ω], with formula 4.1 we obtain the maximum input power $P_{in,max} = -2$ [dBm]. The board also provides an external clock and trigger (synchronization) interface, as well as an HDMI-port which can act as a transceiver, but these functions are not used in our design.

$$P_{in,max} = 10log_{10}(\frac{V^2_{in,RMS}}{R_{in}} \times 1000) \text{ [dBm]} \tag{4.1}$$

The FPGA Mezzanine Card (FMC) interface is defined in the "VITA 57" standard. The FMC125 is a High Pin-Count (HPC) FMC, which allows it to be connected to the ML605 FPGA development board using the HPC-interface, which supports up to 400 pins. We use this port as a Low-Voltage Differential Signal (LVDS)-interface, which provides up to 1 [Gb/s] throughput per pin pair. For the full interface the FMC125 requires 4 ports × 8 bit/Sample × 5 GSample/sec = 160 [G/sec]. This requires 4 ports ×8 bit/Sample = 32 parallel connections running at 1.25 [GHz]. With the 1:2 DMUX-function FMCs, where 2 bytes are sent in parallel, 2 Samples ×4 ports ×8 bits per Sample = 64 pins running at 625 [MHz]. This DMUX-function is used in the design because the hardware of the FPGA can not run at higher speeds. With each channel the ADC passes the clock to the FPGA for reception of the bytes.

The FMC125 board is controlled via an I^2C-bus which is incorporated in the HPC connector. This connection allows interfacing with the peripherals on the FMC125-board. The I^2C bus allows direct access to the Electrically Erasable Programmable ROM (EEPROM) [58] and the temperature sensor [59]. An I^2C-Serial Peripheral Interface (SPI)-bridge [60] is used to access the registers of the clock-generator chip [61] and the ADC for configuration and calibration purposes.

4.1.3 FPGA: ML605

We will use an ML605 [62] development board which contains a Virtex-6 FPGA[1]. The use of an FPGA in an FSR provides the required processing power and is highly reconfigurable logic which enables the reception of signals at different frequency bands.

The FMC-daughter board connects to the HPC-connector of the FPGA-board. The HPC-connector includes an I^2C interface for a 2-way communication with the daughterboard.

The ML605 uses the 10/100/1000 Tri-Speed Ethernet interface to send the data to the PC. This connection is used to control the FPGA program and send data to the PC.

The FPGA uses the LVDS pins to receive the data from the ADC. These pins each provide a maximum data rate of 1 [Gb/s], which means it can only transfer the data from the ADC-board when it is multiplexed; the data rate is 625 [Mb/s] per input. The FMC specifications require that the clock must be included in the interface. Therefore, all four channels (with multiplexed data) have their own clock signal.

The reference design for the FPGA uses 10/100 [Mbit] Ethernet. With a limited time available for the design, we decided we did not need to use the gigabit Ethernet connection. Together with the minimum sampling rate this puts bounds on the required data-rate reduction in the FPGA.

Because the ADC produces samples at a rate of 1.25 [GS/s] and Ethernet only handles 100 [Mb/s] $\approx$ 12 [MB/s], this requires a data reduction factor of at least $\frac{1.25e9}{10e6} = 125$.

4.2 Test Equipment

In this section we introduce the hardware that we used to test our receiver. To create the signals we use GNU-Radio Companion (GRC) on a PC connected to an Universal Software Radio Peripheral (USRP), with a Low-Frequency Transmit (LFTX) daughterboard, to create the analog signals. These signals are then sent to our Vector Signal Generator (VSG) to be mixed to the desired frequency.

[1] FPGA type: XC6VLX240T-1FFG1156

4.2.1 Universal Software Radio Peripheral

To transform the digital signal, generated by a PC, to the analog domain we use an USRP [63]. The USRP connects via Universal Serial Bus (USB) to the PC so the data rate is limited to 8 [MS/s]. For an Frequency Modulation (FM) signal with a deviation of 75 [kHz] we have a total (complex) bandwidth of 150 [kHz]. Because of Nyquist we require to sample at least at twice the highest frequency, requiring a sample rate of over 300 [kHz]. This is well below the maximum sample-rate of the USB connection.

An LFTX daughterboard is connected to the USRP which we use to create an analog complex signal (I/Q-pair). The LFTX-specifications state that its bandwidth is 33 [MHz], but we will only be using the baseband signal generated with the PC which is not more than the bandwidth of the USB port.

4.2.2 Vector Signal Generator

To translate our signals to the desired Radio Frequency (RF) we use the SMBV100A VSG [64]. It allows us to put our signals on a Carrier Frequency (f_c) between 9 [kHz] and 6 [GHz] with a signal strength between -145 [dBm] and $+18$ [dBm]. The maximum bandwidth of the provided I/Q signal may be 500 [MHz], which is well above the width of our test signal.

4.3 Trade-Offs

In this section we describe the trade-offs we had to make due to the limitations imposed on our system by hardware and time. Normally decimation is used for the data rate reduction, but for our project we found that implementing digital filtering in the FPGA was impossible. Therefore we implemented down-sampling without filtering, which causes signal aliasing. We cover the main problems of this phenomena and show how we use it in our design.

4.3.1 Aliasing Explained

Aliasing appears with all digital signals and is caused by the discrete representation. It results in the frequency spectrum repeating every multiple of sample frequency (f_s). The effect can be observed in figure 4.4 where some aliases of the signal are indicated but actually it goes on infinitely. Usually digital signal processing is only performed in the baseband (= 1st Nyquist-zone) and aliasing is not considered a problem. But when signals from outside the baseband are also sampled, the aliases can interfere with each other (at baseband) when they overlap.

Bandpass receivers actively make use of this effect by sampling at a lower rate than twice the highest frequency. There are some considerations and

trade-offs to be made when sub- or down-sampling. Too little subsampling means inefficient use of the available resources, so maximum subsampling is desired. However, using the aliases of signals when sub- or down-sampling has its limits because of the minimum required bandwidth.

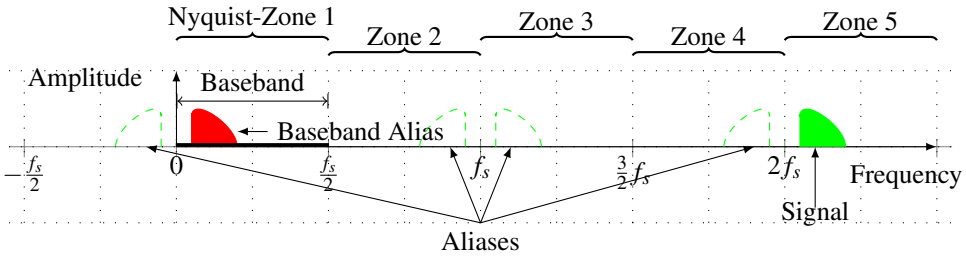
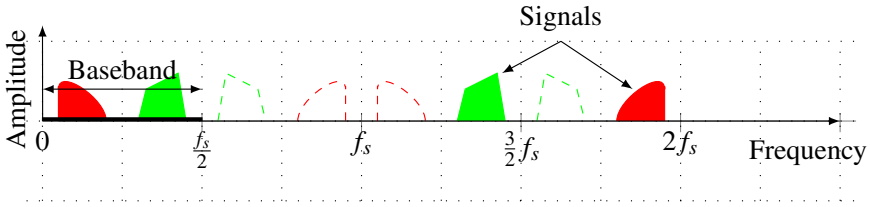


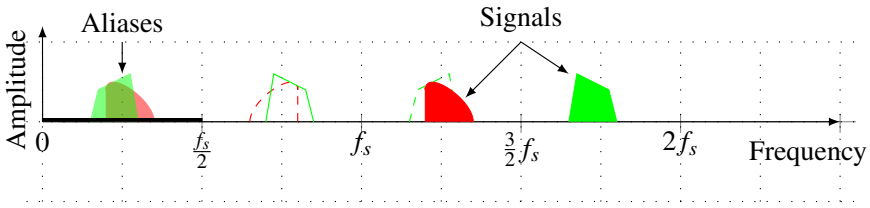
Figure 4.4: Aliases of a baseband signal

4.3.2 Folding

We show what happens if two frequency bands are received by the system but fold over each other. Take two signal bands in the spectrum as in figure 4.5(a). What is clear is that the baseband aliases are separated. Now when we take two arbitrary signals like in figure 4.5(b) it can go wrong. Here the baseband aliases of both signals fold on top of each other, rendering them both useless [32].



(a) Correct signal folding.

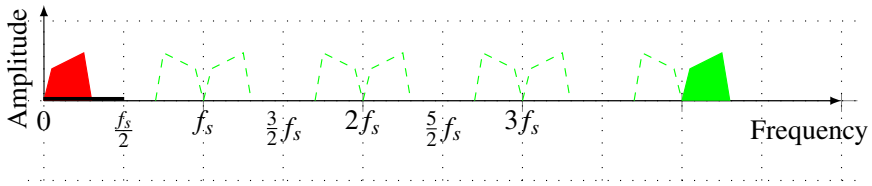


(b) Aliases folding on top of each other.

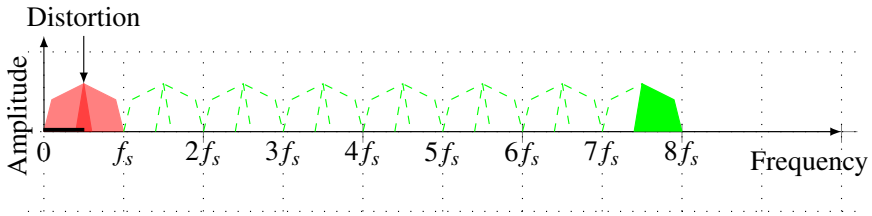
Figure 4.5: Frequency band planning.

4.3.3 Nyquist Bandwidth

Too much subsampling will result in the Nyquist band being violated, which is very bad. This effect is demonstrated by adjusting the scale on the frequency axis accordingly. The signal in 4.6(a) is the signal with an appropriate sampling frequency showing its alias in the baseband. In 4.6(b) we applied too much subsampling, violating the Nyquist-bandwidth. As observed, the signal aliases interferes with the baseband signal (or alias), resulting in the loss of information.



(a) Correct amount of subsampling.



(b) Too much subsampling causing signal distortion at baseband.

Figure 4.6: Different subsampling rates.

4.3.4 Noise Folding

Theory [30] describes that just like every signal, noise will fold to baseband if it is not removed by filtering. The expected Signal-to-Noise Ratio (SNR)-degradation is described by equation 4.2. This phenomena is graphically presented in figure 4.7[2]. In this picture, f_s would be around 400 [kHz], which is two times more than our maximum FM signal (180 [kHz]).

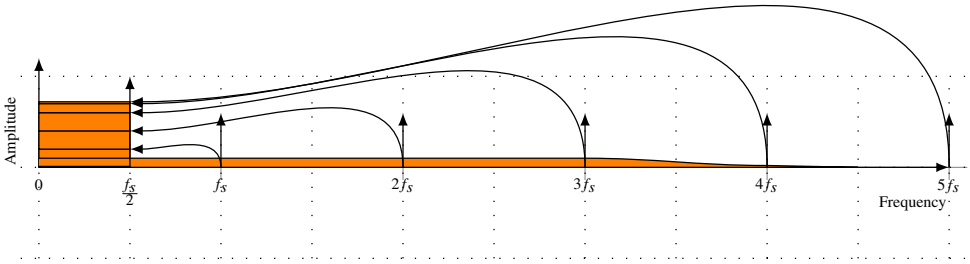


Figure 4.7: Graphic illustration of noise folding to baseband [7].

$$D_{\text{SNR}} = 10log_{10}(n) \text{ [dB]} \tag{4.2}$$

This is 3125× smaller than the input bandwidth of 2 [GHz]. According to 4.2 this should result in a SNR degradation of $10log_{10}(3125) \approx 35$ [dB] due to down-sampling. However, the signal power is unaffected by filtering so it may pass through the subsampling-block without being attenuated, meaning the signal-power remains the same. With no filtering, it can not be that To be able to do that, we first distinguish the different sources of noise in our receiver:

- RF-noise in the signal before the signal enters the ADC caused by atmospheric and environmental effects, the design has no influence on this.
- Clock jitter of both the Sample-and-Hold (S/H) circuit and the sampler causing respectively aperture jitter and sampling clock jitter.
- Thermal noise caused by the physical properties of electrons.
- Quantization noise caused by the discretization of a continuous valued signal.

[2]The noise after the sixth Nyquist zone deteriorates to express the input-bandwidth of the receiver, note that signals above this frequency will also be attenuated.

Quantization Noise

We can assume that quantization noise is uniformly distributed in the spectrum [44]. Quantization will not reduce the SNR when subsampling, because the quantization noise of a 1 [Hz] sampler has the same SNR as a 1 [GHz] ADC. The quantization error is directly linked to the resolution of the ADC: $20log_{10}(2^N) = 48$ [dB]. Following the fact that he ADC can receive a maximum signal strength of -2 [dBm], any signal under -50 [dBm] at the input of the ADC will be lost due to its quantization noise.

Aperture and Clock Jitter

An ideal clock-circuit would produce a perfect pulse or square wave but physical constraints prevent that. The effect is observed in figure 4.8 where a variation in the sample time (Δt) results in a variable sample value($\Delta s(t)$). The variation in the clock is called the clock jitter and the effect it causes is

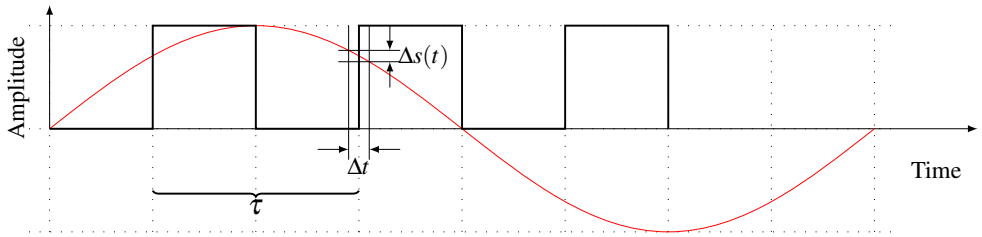


Figure 4.8: Clock and aperture jitter [8].

referred to as phase noise. The Signal to Jitter Noise Ratio (SJNR) is dependent on the maximum clock-jitter (t_{cj}) and the maximum frequency ($f_{\max}$) SJNR $= 20log\frac{1}{2\pi * t_{cj} \times f_{\max}}$; where t_j is the maximum clock jitter and $f_{\max}$ the maximum input frequency.

$$t_j = \sqrt{(t_{cj})^2 + (t_{aj})^2} \tag{4.3}$$

According to [65]:"Since the aperture-time variations are random, these voltage errors behave like a random noise source"; thus the SJNR is the maximum random white noise that's generated by clock jitter. In [66] we find that the total jitter t_{j} is the combination of the clock jitter (t_{cj}) caused by the clocking circuit and the aperture jitter in the ADC(t_{aj}), equation 4.3. Aperture jitter is the variation (uncertainty) in time between two samples, expressed as τ in figure 4.8. In the data sheet of the ADC [26] we find the following values: $t_{aj} \leq 300$ [fs] and $t_{cj} \leq 500$ [fs] resulting in a total $t_j \leq 583$ [fs]. With $f_{\max} = 2$ [GHz] (caused by the bandwidth of the S/H circuit) this results in a SJNR ≤ 42.7 [dBc]. This noise is concentrated on and around the signals [8]. However, it may be hard to observe in measurements because this value is the

maximum value at the cutoff-frequency, where the signal is already attenuated by 3 [dB].

Thermal Noise

Thermal noise is noise caused by the motion of electrons in the system. The total thermal noise power (N_T) is calculated by using the input impedance of a system using equation 4.4. Boltzman's constant $k_B = 1.380,648,8 \times 10^{-23}$ [J/K], the normalized antenna temperature (T) is generally taken at T = 293 [K] and our the bandwidth B = 2 [GHz]. This results in a $N_T = -81$ [dBm].

$$N_T = 10log_{10}(k_B TB \times 1000) \tag{4.4}$$

4.4 Expectations

In this section we combine the theory from chapter 3 with the trade-offs to get a complete overview of effects and limitations of the system that we created. To present the complete system there are considerations concerning the signal power levels in the different stages. First we regard the individual parts of the receiver to quantify the minimum and maximum signal strengths. We also discuss the required signal power from the sending side by regarding the losses caused by the air-transmission and the performance of the receiver system.

4.4.1 Signal Strength Expectations

The minimum signal strength that can be received by any receiver is defined by its thermal noise floor. For our receiver this is set by the input bandwidth of the ADC (2 [GHz]) which results in a thermal noise floor at the antenna of −81 [dBm]. This thermal noise is increased by the noise-factor (NF) of the receiver. For cascaded components the total noise-factor (NF_T) of a receiver is calculated with equation 3.4 where we use the linear numerical-magnitude representation of the gain (G) and NF (F). The NF of the Low-Noise Amplifier (LNA) is specified in the data sheet; the NF of the ADC is calculated using 3.5. Our receiver has a total noise-factor $NF_T \approx 10$ [dB].

	P_{max}	P_{min}	Gain	G	NF	F
TX	20 [dBm]	0 [dBm]	-	-	-	
Antenna	-	-	6 [dBi]	-	-174 $[\frac{dBm}{Hz}]$	
Attenuator	27 [dBm]	0 [dBm]	-30 [dB]	$10^{-3}s$	-	
LNA	5 [dBm]	-82 [dBm]	25 [dB]	316	4 [dB]	2.5
ADC	-2 [dBm]	-50 [dBm]	0 [dB]	1	34 [dB]	240

Table 4.1: In- and output characteristics for several components.

Another hard-limit is set by the quantization noise floor caused by the resolution of the ADC and is set at -50 [dBm] at the input of the ADC. The baseband processing requires SNR= 10 [dB] so the minimum signal strength at the input of the ADC has to be at least $P_{in,min} = -40$ [dBm] to be above the quantization noise floor. With amplification of 25 [dB], by the LNA, we require $P_{in,min} = -65$ [dB] at the input of the LNA.

For the ADC $P_{in,max} = -2$ [dBm] $= 0$ [dBc]. The Spurious-Free Dynamic-Range (SFDR) of the ADC is 48 [dBc], so the spurs caused by the ADC will have a maximum power of -50 [dBm]. Due to the amplification, the input of the LNA must be 25 [dB] lower, resulting in a maximum input power at the LNA of -27 [dBm].

Some loss may be caused by impedance mismatches or imperfections in the hardware, we can not define this beforehand and measurements will identify the significance. We summarize the characteristics of the components used in our test-systems in table 4.1. This table includes the characteristics of an attenuator we used in tests to carefully select the desired signal power and to prevent damage to components.

In figure 4.9 we display the noise floor (P_N), sensitivity ($P_{FM,min}$), minimum (P_{min}) and maximum signal (P_{max}) strength in the receiver chain. From it we can see that the signal and noise power do not change after being digitized by the ADC because we do not apply any filtering in our system. It is also clear that our receiver will not be able to see the thermal noise floor because it lies below the sensitivity.

4.4.2 Signal Transmission

To describe the transmission losses and gains we use figure 4.10. We use GRC with an USRP and LFTX daughter board to create a single sine wave, multiple tones and FM signals. The VSG is used to mix the signal generated by the USRP to the desired f_c. At the receiving side the first component after the antenna is the LNA, for which P_{min} and P_{max} have been defined. The antennas each have +6 [dBi] gain.We distinguish three different ranges between the antennas: one, two and four meters. The signal path-loss over a length d, is calculated with equation 4.5, where $\lambda = \frac{c}{f_c}$ with c the speed of light. This results in respectively 31, 37 and 43 ± 0.5 [dB] loss when $f_]c = 850 \pm 50$ [MHz].

$$P_{\text{path-loss}} = 20log_{10}(\frac{4\pi d}{\lambda})\ [dB] \tag{4.5}$$

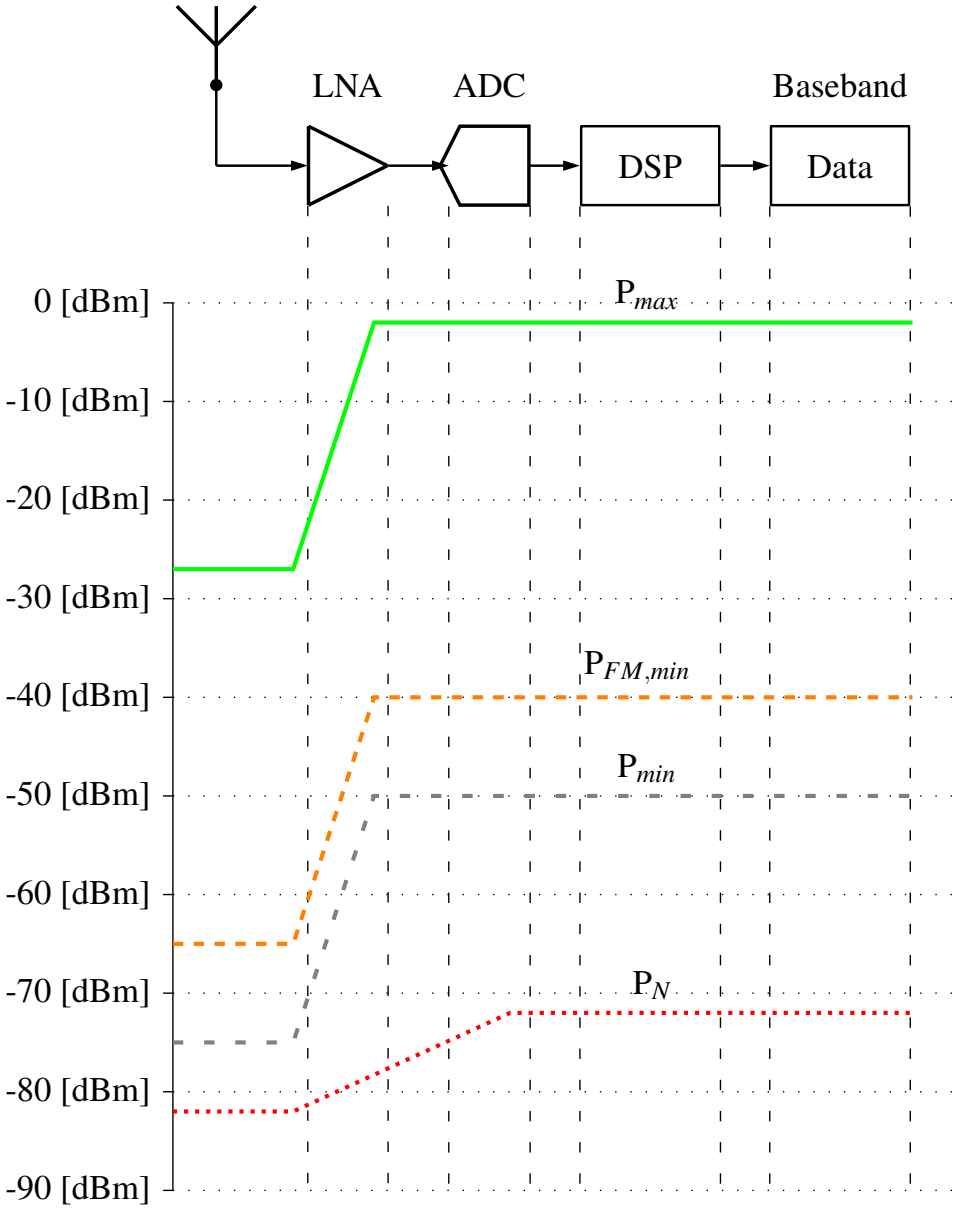


Figure 4.9: Graphical representation of the SNR in our receiver.

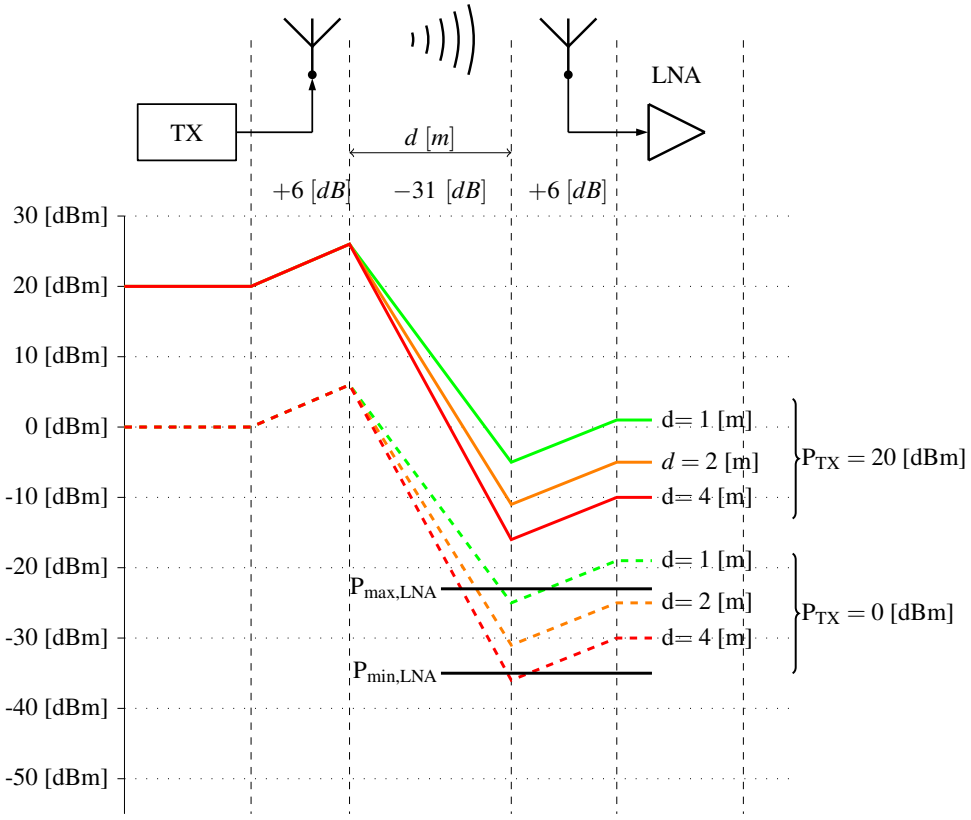


Figure 4.10: Graphical representation of the RF-transmission characteristics.

4.5 Realization

In this section we cover the realization of our FSR system. An evaluation program is provided by 4DSP and is used as the starting point for the development of the system.

4.5.1 Software Operation

The software consists of two parts. The first part is a modified version of the evaluation program provided by 4DSP, made to work under Linux. This host program uses Ethernet to communicate with the FPGA board. When running, the program puts received data in a First-In First-Out (FIFO) buffer; this buffer is then read by the back-end program which is implemented using GRC [67]. When running the software the down-sampling factor of the FPGA must be set in the GRC-program. Also the GRC-program must be started before the server program, because otherwise that one will fill the FIFO buffer too fast, which causes errors.

Hardware Controller Program

On the PC a host program provides the control of the FPGA hardware. It was originally written for use under Windows by 4DSP, but some alterations were made. The host program tells the FPGA (via Ethernet) what to do. It starts by detecting if the FPGA and ADC are connected, then it initiates the calibration of the clocks and the ADC and finally start the capturing of data. When the FPGA has started to send data, the Ethernet connection will become useless for other functions as the data-stream is usually too high.

The original Ethernet library from 4DSP has been modified to work on Linux. It uses RAW Ethernet frames with a custom protocol which is defined in the Stellar-IP documentation: SD040 MAC Engine Star. An Ethernet frame consists of:

- 6 bytes destination MAC address
- 6 bytes source MAC address
- 2 bytes protocol (0xF000)
- 46-1500 bytes data
- 4 byte checksum (Not used)

Using RAW packets reduces the required overhead compared to standard TCP and UDP packets. The only means to receive this kind of packets under Linux is to listen directly on the Ethernet port. We make use of Wireshark, which keeps the port opened for our program to read the received packets.

GNU-Radio Companion Digital-Signal Processing Program

GRC provides a digital signal processing-tool on the PC to provide the back-end of our receiver chain. With GRC we can perform the required baseband processing. Especially, but not limited to, the filtering and demodulating of FM signals.

The subsampled data from the ADC, received via Ethernet, is put into a FIFO by the host program and read by the GRC-program. We present the setup of the FM-receiver in figure 4.11. In this example the received signal is mixed to make sure it ends up around Direct Current (DC) before demodulating it. The resampling is applied to match the sample rate of the sound card of the PC. The raw data consists of 8-bit samples ranging from 0

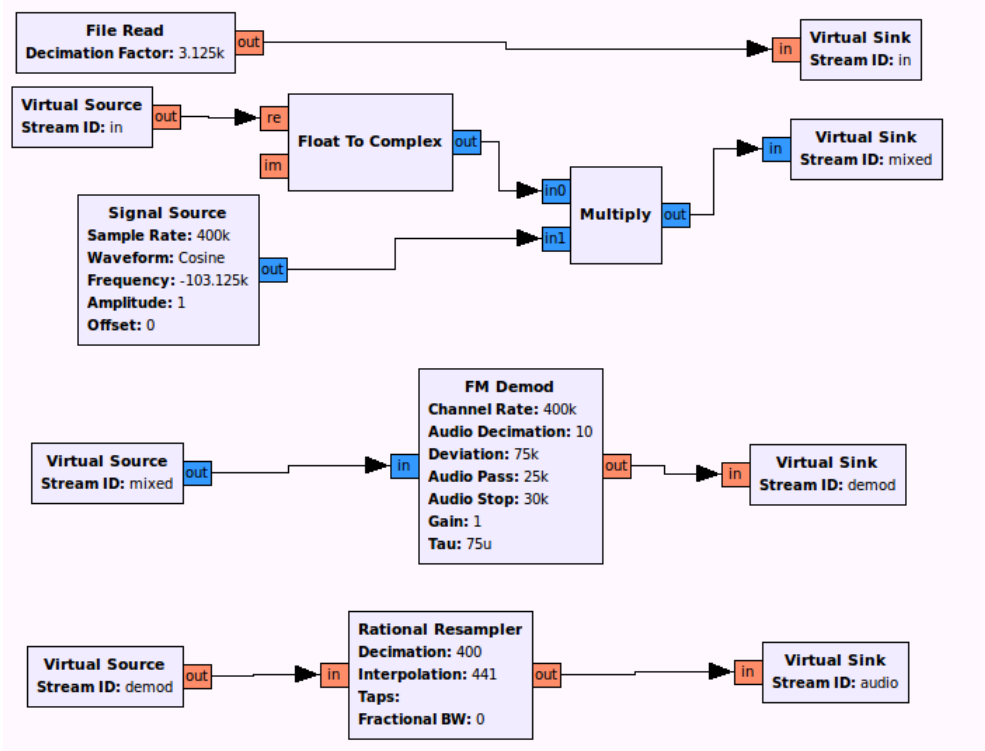


Figure 4.11: GRC FM-Receiver implementation.

to 255 and therefore we apply some preconditioning-conditioning in GRC to work with these samples. We found that a sample-value of 126 corresponds with an input of 0 [V], so we subtract that amount from the original sample to decouple the DC from our signal. In GNU-Radio a signal with an amplitude of 1 [$V_{p\text{-}p}$] corresponds to 0 [dB], therefore it makes sense to scale our signal accordingly. The values, which range from -126 to 129 after the subtraction, are divided by 127 so 1 bit now corresponds to a value of 7.87 [mV] in GNU-Radio. This is all done inside the ˜File Read˝-block in figure 4.11.

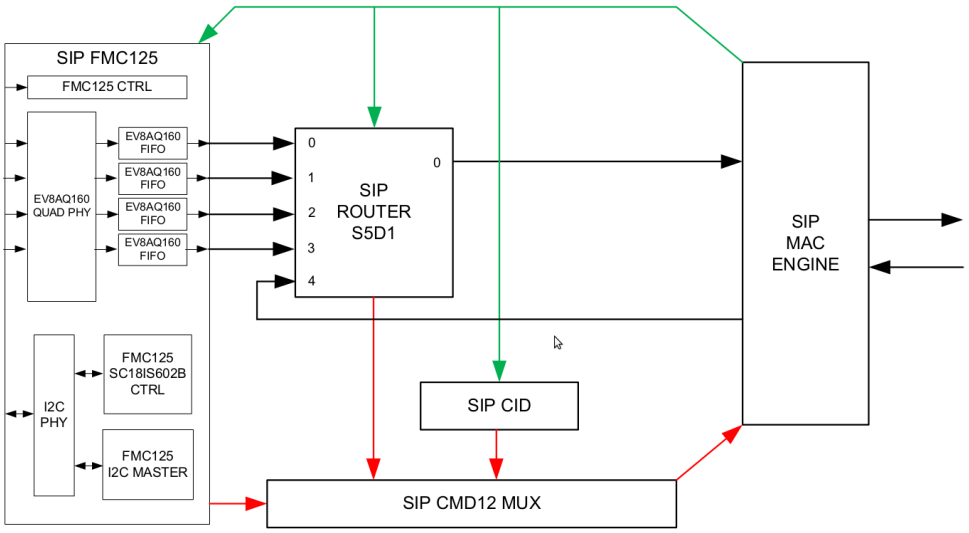


Figure 4.12: Firmware design for the ML605 with an FMC125 from [9].

Due to the offset we lose a value at the lower edge of ADC range; the maximum input voltage of the ADC potentially provides values corresponding to values over 0 [dB]. However, working with signals at the full-scale of the ADC is risky because the signal could be clipping and cause unwanted effects.

4.5.2 Hardware Implementation

A hardware configuration is provided for the ML605 FPGA board, by 4DSP, which responds to commands send to the board via Ethernet from the host program. The IP-blocks use a communication protocol designed by 4DSP, based on their Stellar IP tool, to communicate between them, forcing additional blocks that are created to adhere to this interface. New blocks are therefore best created using their (Windows-only) Stellar IP tool. The configuration contains the IP-blocks written by 4DSP; an overview of the reference design is shown in figure 4.12. The ports on the left side of the image are connected to the FMC125 ADC; to the right is the Ethernet connection to the host-program running on the PC.

Subsampling

We implemented a down-sampling block called:"decimator", see appendix 5.1. Because of the high data-rate from the ADC we place this close to the input. With the Stellar IP tool we place the down-sampling between the "EV8AQ160 QUAD PHY" and the corresponding "EV8AQ160 FIFO" blocks. The decimator-block performs at least a subsampling of 8 with no practical limit on the highest factor. This factor must be decided when synthesizing

the design in the Xilinx tools. It also provides the signals to and from the FIFO and the PHY because they use different clock-domains now that the data is downsampled. The original hardware uses 16 bits per sample, in our implementation we convert this back to 8 bits per sample, effectively reducing the eventual data rate.

4.6 FFTs for Measurements

In this section we elaborate the information about the signals that we obtain from the Fast-Fourier Transform (FFT)-images produced by GRC, because we will use it for our results.

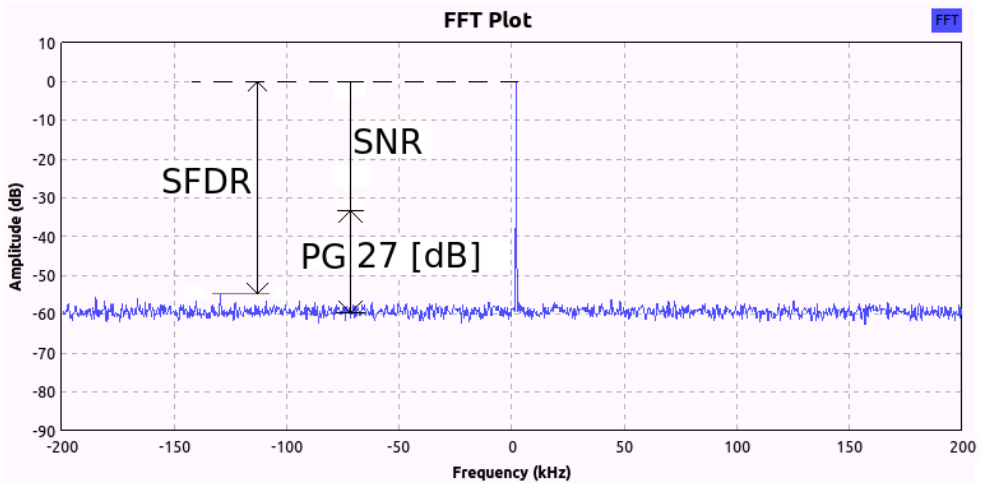


Figure 4.13: Overview of SNR and SFDR when using the FFT.

We can define the noise power by taking the average noise-level from figure 4.13. We have to add the FFT process-gain for using a 1024-point FFT, so we obtain $\mathrm{PG_{FFT}} \approx 27$ [dB] which results in the effective noise-power P_{noise} 4.6 [68].

$$\mathrm{PG_{FFT}} = 10log_{10}\left(\frac{\mathrm{M}}{2}\right) \text{ [dB]} \tag{4.6}$$

To prevent spectral leakage we could use specific windows [69]. For FM, and other modulated signals, the Blackman window may be preferable [70]. We use a rectangular (boxcar) window because this does not alter the strength of the signal; which is useful when comparing signals in different FFTs with each other. The spectral leakage and poor amplitude accuracy, which are a trade-off for this window, will have to be prevented by carefully selecting the frequencies. If this is performed correctly, the signal strength (P_{signal}) will be visible in the FFT.

The SFDR we use is expressed relative to the carrier: it is the difference between the signal power and the strength of the highest spur in the FFT[3]. Because the spurs are multiples of the frequencies that we use, we expect them to fall in their appropriate frequency bins. However, if this is not the case, currently we have no solution to prevent this with our method.

4.7 Evaluation

In this section we evaluate the results from the FSR-setup that we created. We perform measurements with a single tone, two tones and an frequency-modulated signal; the signals are modulated to different carrier frequencies. We analyzed the following signal properties:

- Minimum and maximum signal power
- Phase noise
- Inter-Modulation Distortion (IMD)
- FM signal audio quality.

With each of these measurements we observe the signal power, the noise power and the power of the highest spur, usually the IMD3. This allows us to evaluate the performance of the receiver by verifying its SNR and the SFDR.

First we use a wired connection between the Transmit (TX) and Receive (RX) side, so we can accurately condition the signals that are fed to the ADC. This allows us to verify the theory about the performance of the receiver without atmospheric or other influences. In the second part we verify the wireless receiver where the LNA and antennas will have an impact on the signal.

4.7.1 Wired Receiver

We performed the initial experiments with a wired connection because the system is much less susceptible to interferers allowing us to make. I also allowed us to carefully define the minimum and maximum signal strengths fed to the ADC with our VSG. The two degrees of freedom we take for the transmitter are: Signal strengths and carrier frequency. For the frequency modulation in the corresponding measurement, we take at the conventional 75 [kHz].

[3]In some literature the SFDR is relative to (1 [dB] below) the full-scale.

Single tone Results

To evaluate the ADC we performed a series of measurements with single tones to determine what the minimum and maximum signals are at certain frequencies. This measurement can provide insight in the aperture jitter of the signal by comparing the SNR of the signal at different frequencies [71]. It also shows how the data from the ADC corresponds with the analog signal.

We use the USRP with a LFTX board to generate a single tone frequency and then use the VSG to mix it to the desired carrier frequency. We used 3 different carriers for the signals, they are chosen such that they can later be used in the wireless system as well[4]. The 781 [kHz] is a relatively low frequency, slightly more than ten times the FM-deviation of 75 [kHz]. The 100.1 [MHz] signal is chosen because its situated in the commercial FM-band. The 1.8 [GHz] signal is close to the maximum analog bandwidth of the S/H circuit, 2 [GHz], note that with $f_s = 1.25$ [GHz] we are applying subsampling here. We present for different frequencies the transmitted power ($\mathrm{P_{TX}}$), the received power ($\mathrm{P_{RX}}$), the SNR and the SFDR in the tables 4.2, 4.3 and 4.4. With $\mathrm{P_{TX}} = -10$ [dBm], we measured $\mathrm{P_{RX}}$ at higher frequencies and obtained the values presented in table 4.5.

P_{TX} [dBm]	$\mathrm{P_{RX}}$ [dBm]	SNR [dB]	SFDR [dB]
-1	-7	48	43
-10	-16	39	48
-40	-46	9	23
-50	-59	-4	10

Table 4.2: f_c= 781[kHz]

P_{TX} [dBm]	P_{RX} [dBm]	SNR [dB]	SFDR [dB]
-1	-7	47	49
-10	-15	39	51
-40	-46	9	22
-50	-58	-4	12

Table 4.3: f_c= 100.1[MHz]

P_{TX} [dBm]	P_{RX} [dBm]	SNR [dB]	SFDR [dB]
-1	-14	39	41
-10	-21	32	44
-40	-58	0	2

Table 4.4: f_c= 1.8[GHz]

[4]Due to the antenna bandwidth the frequencies could not be chosen too far below 800 [MHz].

f_c	P_{RX}
2 [GHz]	−24 [dBm]
3 [GHz]	−30 [dBm]
4 [GHz]	−37 [dBm]
5 [GHz]	−46 [dBm]
6 [GHz]	−48 [dBm]

Table 4.5: f_c= 1.8[GHz]

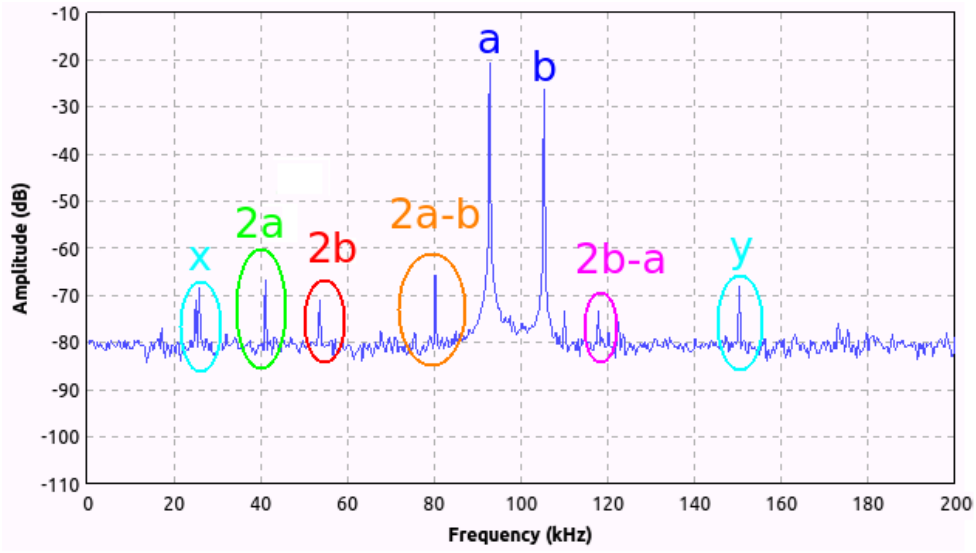


Figure 4.14: FFT of a dual-tone signal at 400 [MHz] showing the IMD.

Dual Tone Signal

For the dual-tone measurement we took a static value for the sine wave frequencies. In figure 4.14 we labeled the signals and spurs that we observed. Signal ″a″ and ″b″ are the created signals. Both signals are mixed up by the VSG to an IF, for the test-case in the image we took 400 [MHz]. We made the ″b″ signal slightly weaker than the ″a″ signal to show the difference in power levels and the resulting spur-powers. The sine waves are half the full-scale in amplitude (−6 [dB]) to prevent clipping and the signal power $P_{TX} = -10$ [dBm]. We see two spurs very close to each other at ″x″, the rest of the spurs are indicated with the combination of signals that causes them.

FM Audio

Tests show that $SNR_{audio} \geq 40$ [dB] results in good, uninterrupted sound between 200 and 1000 [Hz]. The FM-signals were received and decoded prop-

erly when $P_{\mathrm{TX}} \geq -35$ [dBm]. The transmitting power could not exceed $\mathrm{P}_{\mathrm{TX,max}} = -1$ [dBm] without clipping. This results in a Dynamic Range (DR) for the FM-reception of the ADC of 34 [dB], which is as expected.

Observations

An initial offset exists between the transmitted and received signal power. The ADC is should accept signals up to -2 [dBm], the transmitter sends a -1[dBm] signal and the received signal power is -7 [dBm]. If we take the transmitter to be perfect, a 1 [dB] loss in the cable explains that the maximum transmitter power can be 1 [dBm]. The -5 [dBm] that is lost is an imperfection in the measurement and probably caused by the scaling performed in GRC. However, the SNR and SFDR are relations between the signals and the received signal strength decreases correctly with the transmitted power, therefore the absolute value is not important to obtain a correct result.

The IMD of the signals are very clear under 45 [dBc] $= -65$ [dBm]. The IMD3 *2b-a* results in a lower power than this, which is expected because the IMD3 attenuates (or amplifies) at a higher rate than the IMD2. We observe that the IMDs fall well below the noise-power level of $\mathrm{P}_{\mathrm{noise}} = -53$ [dBm].

The f_c appeared to be not as stable as expected which caused it to vary continuously. This effect increased as the f_c got higher, making it harder to make the signal to fall in a single frequency-bin of the FFT. Therefore it is possible that some of the phase noise that we expected around the signal may be caused by the spectral leakage of the FFT. This makes it impossible for us to say anything about the phase-noise in the measurement.

During testing we found a bit inversion of the sixth bit in a sample; the value is x3F when it should have been x7F. It does not occur for other values, therefore it may be a timing or connection problem on the hardware platform, FPGA or ADC. The rate at which it occurs is low (1 in every 50,000 samples max, but usually much lower). Because the FFT-size is 1024 (consecutive) samples, our measurements can be used as long as a measurement does not contain a faulty sample[5].

The VSG has two modes to choose from: ″Low Noise″ and ″Low Distortions″. The second mode has a considerable influence on the spur-strength; spurs are decreased by about 10 [dB] compared to the first. This leads to the expectation that part of the SFDR is on the account of the transmitter. In figure 4.14 this distortion is visible at signal ″x″.

The ADC introduces two distortions that occur even when TX is not connected. Unconnected these occur below -53 [dBm], when connected they are at most -47 [dBc]. While not an issue, they are remarkable, in figure 4.14 they occur at ″x″ and ″y″.

[5] A faulty sample may be considered as an impulse, containing many frequencies, which would cause a much higher noise floor.

4.7.2 Wireless FM-Radio

To verify the theory and test our receiver capabilities we perform a measurement of the wireless receiver. The setup uses an antenna, LNA and the FSR. We place a Surface Acoustic Wave (SAW) bandpass filter is between the antenna and the LNA to illustrate the beneficial effects of analog filtering and to identify the noise-sources.

Results and Observations

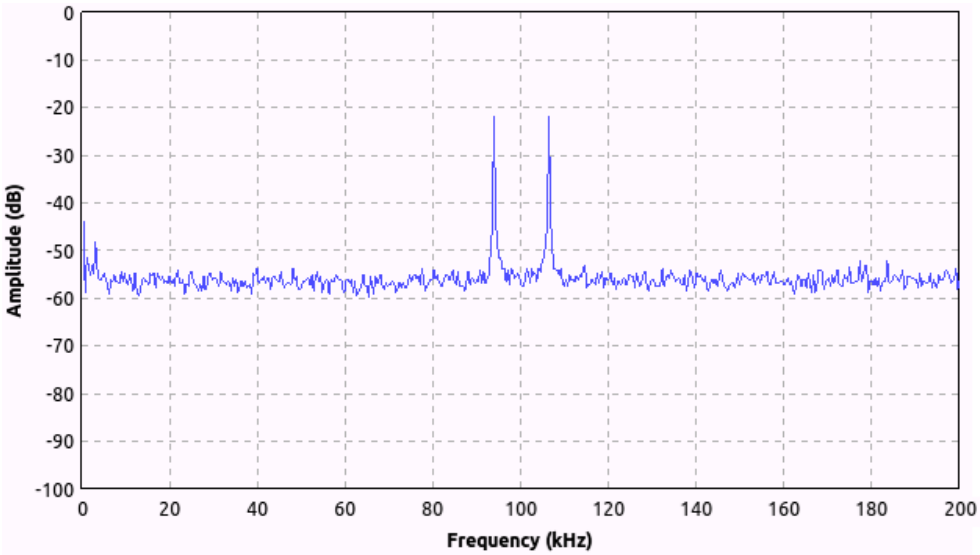


Figure 4.15: FFT of a dual-tone signal at 433 [MHz] via wireless communication.

The noise power of the wireless receiver has increased by 25 [dB] compared to the wired receiver and we obtain the signals as depicted in figure 4.15. This decline in SNR could be the result of interferers from other bands that are received and folded back over the signal. When the SAW is added to the system between the antenna and the LNA we obtain the results as shown in 4.16. Here can be seen a reduction of the noise power of ~ 20 [dB], which is only a deterioration of a few dB compared to the wired receiver. We expect that this is the result of the noise figure of the LNA, which is 4 [dB].

4.8 Conclusions

In this section we presented the FSR system that we used to verify the theory. We observed the effects of downsampling without filtering and defined what

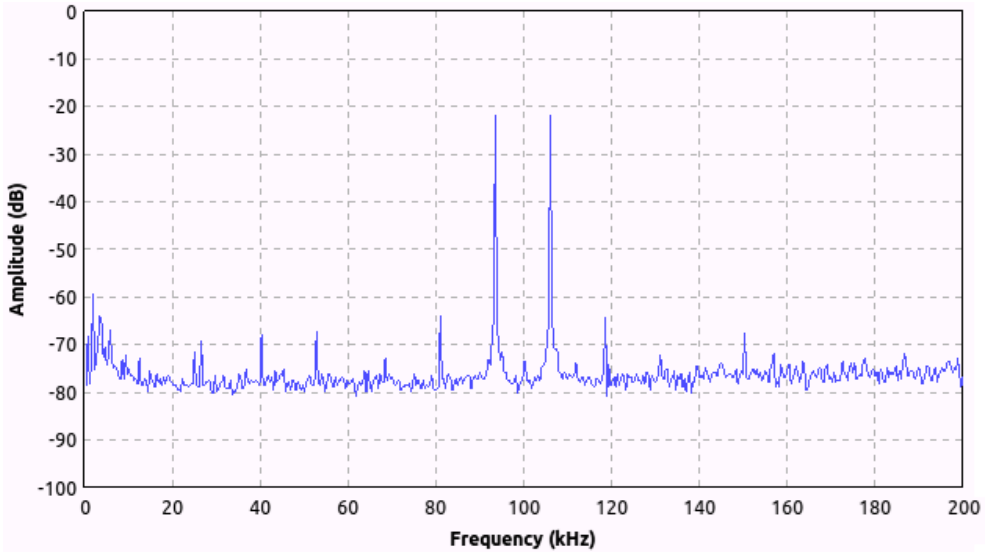


Figure 4.16: FFT of a dual-tone signal at 433 [MHz] via wireless communication using a SAW-filter.

the effect would be. Other signal properties and expectations were expressed before the measurements are presented.

While often referred to as ˜only a few dB˝ , this very quickly become an enormous difference. We verified that the SNR did not degrade in our system conform our expectations in this chapter, but contradicting our initial research in chapter 1.

We were unable to say anything about the phase-noise due to the nature of our measurements. The effect of aperture jitter became apparent with the unstable clocks causing a frequency shift of our signals. The instability of the carrier frequency increased as it became higher, making the transmitter clock the suspected source. However, this could not be visualized in the report, neither can we put a number on it, making it ambiguous to reason about.

The difficulties of working with frequencies in the gigahertz range became apparent. The instability of the clock and bandwidths of the components has a great impact and we had to carefully decide on the used frequencies. Also the sample rate had a great impact on the data rate and the logic in the FPGA, where the possibilities and limits of the hardware are put to the test. This may be the result of the anomaly in our measurements.

Chapter 5

Conclusions

In this thesis we presented our research on the design of Full-Spectrum Receivers (FSRs) and the trade-offs involved. We made an estimation on what the requirements of a practical receiver would be. With the time available for a graduation project we had to make design choices for our own implementation. We presented the implications of these choices on the design. We verified the theory by implementing the design and creating a prototype FSR.

The goal was to implement an FSR with the given hardware. The requirement that we made for our receiver could not be met. The maximum signal strength that the Analog-to-Digital Converter (ADC) can handle is -2 [dBm]. With eight bits the ADC has a best case Signal-to-Noise Ratio (SNR) of 50 [dB]; the datasheet defined SNR $\geq$ 45 [dB] at $f_c = 100$ [MHz] and we measured 47 [dBm]. At $f_c = 1.8$ [GHz] we are subsampling the signal, and obtain SNR = 39 [dB]. This deterioration is caused by the analog bandwidth of the Sample-and-Hold (S/H)-circuit, but is more than the expected 3 [dB] cut-off frequency.

We used a Low-Noise Amplifier (LNA) to be able to receive weak signals and to create a low total Noise Figure (NF) of our receiver using the Friis' equation for cascaded components. The input signals are amplified by 25 [dB] with the LNA. This allows our receiver to discriminate signals as low as -75 [dBm]. With the required 10 [dB] for FM-demodulation, signals at -65 [dBm] may be received, which is 25 [dB] higher than the desired -90 [dBm]. Because of the maximum input power to the ADC and the amplification of the LNA, signals can have a power of -27 [dBm] before clipping occurs. This is 17 [dB] lower than the desired maximum signal power of -10 [dBm]. This means that we are missing a total of $25 + 27 = 52$ [dB] in range for the reception.

We made some trade-offs to the design to fit the project in the time for a graduation assignment. In the realized system we were unable to meet the desired sampling rate of 5 [GHz]. Instead we sample at a rate four times lower, $f_s = 1.25$ [GHz]. Filtering should be implemented when downsampling signals, but the effort for a proper filter in the Field-Programmable Gate Array (FPGA) at these speeds was expected to be too great for this project so it was left out. The implications of these trade-offs are presented, supported by theory, and validated with the prototype FSR.

Due to the bandwidth of the LNA we can only receive signals with car-

rier frequencies between 100 [kHz] and 1 [GHz]. The analog bandwidth of the S/H circuit is wide enough (2 [GHz]), but because sample frequency (f_s) = 1.25 [GHz] we apply subsampling for signals (and noise) above 625 [MHz]. Subsampling and downsampling are identical in both theory and practice. With downsampling explored, we also obtained the knowledge about subsampling allowing us to understand the effects. The absence of filtering greatly reduced the selectivity of our system so we could only receive (FM) signals that are at least 10 [dB] stronger than the strongest interfering signal.

We expected that the application of downsampling without filtering would be very problematic due to folding and aliasing. However, the aliasing of noise in our system turned out not to be the problem we expected it to be. With our wired system we completely avoided the interference from other signals. The noise floor was dominated by the quantization noise, with an SNR of 50 [dB] for the receiver. We were unable to observe the thermal noise with this system, while we initially expected it to appear. In the wireless system the noise-floor was made up of other signals that folded back to baseband.

Besides the noise floor, we also identified Inter-Modulation Distortion (IMD) putting a limit on our receiving capabilities. The IMDs, caused by nonlinearities in the components of the receiver, appeared at the same power level as the noise, -50 [dBm]. We found that for wired systems all interferers were below the sensitivity of our ADC and only spurs were visible in the measurements. Therefore the FSR may be well applied in wired communication systems.

In our wireless receiver, the noise floor before the LNA has a power of -55 [dBm]. Including the 25 [dB] amplification, this resulted in the noise floor at the ADC to be -30 [dBm]. This noise floor is probably caused by an interfering signal which folds back. To verify this, we applied a Surface Acoustic Wave (SAW) Band-Pass Filter (BPF) and found the actual noise-floor of the receiver at -75 [dBm], almost the same as the wired system.

The power of this interferer, -55 [dBm], is much lower than the -10 [dBm] we initially expected as the maximum interfering signal power [16]. This allows us to receive FM signals with a power between -45 and -30 [dBm] without applying filtering in our system.

We can conclude that an FSR is very susceptible to interferers. However the signals found in the air were not so strong that we required the full dynamic range that we initially defined. This makes the FSR very interesting for reception, as long as you can ensure the absence of too strong signals.

5.1 Future work

We were able to receive the test signals with our receiver but there are still improvements possible that could increase the performance of our system. The FSR is very susceptible to interferers, but the interferers were not as strong as expected. We can apply more amplification, in the analog part of our receiver chain, to increase the sensitivity of the receiver. This will allow us to better match the input range of the FSR to commercial signals but will only work if the received signal powers are guaranteed not to exceed the maximum power.

We presented possible implementations for the digital signal processing but due to the time available we were unable to implement them. If we want to capture multiple signals we will have to apply proper filtering to prevent the signals from folding over each other.

While our ADC allows sampling up to 5 [GHz], we only use one fourth of this. Future designs should use the maximum sampling frequency. Together with proper digital filtering, this will prevent the subsampling from potentially destroying the signal before it can be digitized.

The LNA we used has a passband between 100 [kHz] and 1000 [MHz]; a noise-factor of 3 [dB] and 25 $\pm$ 0.5 [dB] gain. These parameters are not bad, but there are better parts available [18] [72]. An improvement to the system would be to take an LNA with a wider band, a lower NF and a flatter passband ripple.

For practical applications of wireless FSRs, the theory describes that we will have to handle the strongest signals -10 [dBm] as well as the weakest -100 [dBm] For that, the ADC in wireless receivers requires at least 13 bits. State-of-the-art ADCs found in [73] and [74] provide an Signal-to-Noise and Distortion Ratio (SNDR) of 61 [dB] ($\approx$ 10 bits) with respectively $f_s = 2.5$ [GHz] and $f_s = 5.4$ [GHz].

Theory on processing gain (averaging) describes that for each two times oversampling rate, half a bit of effective resolution may be gained [75]. With the given f_s we can say that for 3 additional bits (on top of the 10 already available) a sampling rate of 5.4 [GHz] we could service (in theory) a bandwidth of $\frac{2.7}{2^6} = 42$ [MHz] with a Dynamic Range (DR) of 13 [dB]. However, we will have to investigate the effects of IMD when applying processing gain.

In [74] they claim that the ADC consumes only 500 [mW], which is well on its way to be used in mobile devices. However, this does not include the power consumption that is required for the digital processing in the FSR, which will put the total consumption probably too high. Also because FSRs do not match the DR, and thus sensitivity and selectivity, of conventional receivers and standards, we expect they will not be implemented commercially anytime soon.

System manual

This chapter contains requirements and information on how to run and change the presented system for future work. For an overview of the system see the chapter 4.

Requirements

- The system is created on Ubuntu 12.04 LTS, 32-bit.
  64-bit systems are expected not to introduce problems.
- For compiling the GNU-Compiler-Collection (GCC) is used.
  Version 4.6.3 was used during testing.
- Wireshark must be running and listening on the used interface.
  The program uses RAW packets and no packet listener is implemented, Wireshark handles this. Version 1.6.7 was used during testing.
- The bitfiles are generated using Xilinx ISE 13.4.

Note that the program is written for a test-system with a specific ethernet interface: 'eth1'. This parameter would have to be changed accordingly on other systems in *ethapi.c* - line 54.

Running the system

Before running:
Currently the system uses only 1 ADC: channel A. The maximum sample rate is 1250 MB/s. The total downsampling factor can be configured by two 'parameters'in the board. The gnuradio receiver must be configured to comply with this factor. The first factor governs the sampling rate by the ADC and is set in *fmc125_clocktree.c*. Line 128/129 decide if subsampling is used or not. If it is used, line 116-120 decide the subsampling factor. The FPGA should be programmed with a bitfile using the desired decimation. Take into account that total downsampling rate is the decimation $\times$ subsampling factor. *fmc125_clocktree.c* - line 128. Make sure Wireshark is launched and monitoring the desired interface (on the used system this is eth1). Make sure gnuradio-companion is running a receiver as provided, otherwise the ADC and FPGA system will begin sampling, but the sink will fill up really fast and the program will crash.

Running:
The program must be run as administrator because it requires low-level access to the ethernet registers. The provided make-script executes all these steps and runs the program after compiling. Run with *make*, the system will require an administrator password (first time only) and start after it's provided.
Shortcomings:
Quite some digital processing is performed on the PC. If it can't keep up with the datarates, the buffer will fill up faster than it's read and the program will (intentionally) exit. Rarely the filling up of this buffer happens without a reason.
Due to misalignments between the soundcard clock and the ADC clock, the gnuradio program is experiencing a slight underflow. On other systems this may result in a slight overflow, experiencing above mentioned error.
Overview of relevant datasheets and documents:

- FMC12x User Manual [6]
- Quad ADC with 8-bit Resolution, EV8AQ160 [26]
- 12-Output Clock Generator with Integrated 2.8 GHz VCO, AD9517 [61]
- I2C-bus to SPI bridge, SC18IS602B [60]
- Temperature Sensor and 8-Channel ADC, ADT7411 [59]
- 2K I2C Serial EEPROM, 24AA02/24LC02B [58]

4DSP manuals:

- FMC125 Star Quad 8-bit 1.25Gsps ADC Daughter Card
- MAC Engine Star - SD040 (sip_mac_engine).pdf

Downsampler

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.NUMERIC_STD.ALL;
entity decimator is
    Port ( clk : in  STD_LOGIC;
           rst : in  STD_LOGIC;
           en_in : in  STD_LOGIC;
           en_out : out  STD_LOGIC;
           din : in  STD_LOGIC_VECTOR (127 downto 0);
           dout : out  STD_LOGIC_VECTOR (127 downto 0));
end decimator;
architecture Behavioral of decimator is
constant ndec : integer := 10;
signal data : std_logic_vector(127 downto 0) := (others => '0');
constant decimator : std_logic_vector(16 downto 0) :=
                        std_logic_vector(to_unsigned(ndec, 17));
signal counter: std_logic_vector(16 downto 0) := (others => '0');
signal en_cnt : std_logic_vector(7 downto 0) := (others => '0');
begin
assert (ndec >= 8) report "Invalid decimation" severity failure;
dout <= data;
proc: process (clk, rst, din, en_in, en_cnt, counter)
begin
  if (rst = '0') then -- Asynchronous reset
    if rising_edge(clk) then
    if signed(counter) >= signed(decimator) then
      en_cnt <= std_logic_vector(unsigned(en_cnt) + 1);
      CASE counter(2 downto 0) IS
        WHEN  "000"  =>  data(127 downto 120) <= din(7 downto 0);
        WHEN  "001"  =>  data(127 downto 120) <= din(23 downto 16);
        WHEN  "010"  =>  data(127 downto 120) <= din(39 downto 32);
        WHEN  "011"  =>  data(127 downto 120) <= din(55 downto 48);
        WHEN  "100"  =>  data(127 downto 120) <= din(71 downto 64);
        WHEN  "101"  =>  data(127 downto 120) <= din(87 downto 80);
        WHEN  "110"  =>  data(127 downto 120) <= din(103 downto 96);
        WHEN  "111"  =>  data(127 downto 120) <= din(119 downto 112);
        WHEN OTHERS  =>  data(127 downto 120) <= (others => '0');
      END CASE;
      data(119 downto 0) <= data(127 downto 8);
      counter(16 downto 3) <= (3 => '1', others => '0');
      counter(2 downto 0) <= std_logic_vector(unsigned(counter(2 downto 0))
                                        + unsigned(decimator(2 downto 0)));
    else
      counter <= std_logic_vector(unsigned(counter) + 8);
    end if;
    if (unsigned(en_cnt) = 16) then
      en_cnt <= (others => '0');
      en_out <= en_in;
    else
      en_out <= '0';
    end if;
    end if;
  else
    counter <= (others => '0');
    en_out <= '0';
    data <= (others => '0');
  end if;
end process;
end Behavioral;
```

Bibliography

[1] P. Cruz, H. Gomes, and N. Carvalho, "Receiver Front-End Architectures - Analysis and Evaluation," pp. 495–521, 2010.

[2] B. Murmann, "ADC Performance Survey 1997-2013," 2013.

[3] M. Patel and P. Lane, "Comparison of downconversion techniques for software radio," *Proc. of LCS 2000*, pp. 2–5, 2000.

[4] L. Der, "Frequency Modulation (FM) Tutorial," *Silicon Laboratories Inc*, 2008.

[5] J. Meyer and S. Menzel, "Ultra High Speed Digital Down Converter Design for Virtex-6 FPGAs," *Proceedings of 17th International OFDM Workshop*, no. Ddc, 2012.

[6] 4DSP, "FMC122 / FMC125 / FMC126 User Manual," 2010.

[7] S. Karvonen, "A low noise quadrature subsampling mixer," *The 2001 IEEE International Symposium on Circuits and Systems*, pp. 790–793, 2001.

[8] V. Arkesteijn, "Jitter requirements of the sampling clock in software radio receivers," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 53, no. 2, pp. 90–94, 2006.

[9] 4DSP, "FMC125 Reference Firmware," 2010.

[10] J. Happich, "RF Engines pushes its wideband digital down converters to precess 1 GHz bandwidth," *Electronic Engineering Times Europe*, 2010.

[11] ECC, "The European Table Of Frequency Allocations And Applications In the Frequency Range 8.3 kHz to 3000 GHz," 2013.

[12] A. A. Abidi, "The Path to the Software-Defined Radio Receiver," *IEEE Journal of Solid-State Circuits*, vol. 42, pp. 954–966, May 2007.

[13] O. Jamin and V. Rambeau, "An RF front-end for multi-channel direct RF sampling cable receivers," *Proceedings of the ESSCIRC (ESSCIRC)*, pp. 347–350, Sept. 2011.

[14] Broadcom, "White Paper: Full-Band Capture Cable Digital Tuning," 2011.

[15] D. Wang and Q. Rao, "Integrated Design of Multiple Antennas for WiFi/Bluetooth/GPS Mobile Communication," *Session 3A3 Antennas and Array: Theory and Design 2*, pp. 755–758, 2010.

[16] G. Watkins, "Potential interfering signals in software defined radio," *6th IEEE High Frequency Postgraduate Student Colloquium*, pp. 41–46, 2001.

[17] J. Geier, "Wi-Fi: Define Minimum SNR Values for Signal Coverage," 2008.

[18] L. Lolis, "Agile bandpass sampling RF receivers for low power applications," 2011.

[19] D. Akos and J. Tsui, "Design and implementation of a direct digitization GPS receiver front end," *Microwave Theory and Techniques*, vol. 44, no. 12, pp. 2334–2339, 1996.

[20] B. Daya, "Super-heterodyne FM Receiver Design and Simulation," 2006.

[21] Mini-Circuits, "Low Noise Amplifier - ZFL-1000LN Datasheet."

[22] C. De Vries and R. Mason, "Subsampling Architecture for Low Power Receivers," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 55, pp. 304–308, Apr. 2008.

[23] B. Jonsson, "Using Figures of Merit to Evaluate Measured A/D-Converter Performance," *Proc. of 2011 IMEKO IWADC & IEEE ADC Forum*, no. 1, 2011.

[24] T. Sundstrom, "Power dissipation bounds for high-speed Nyquist analog-to-digital converters," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 56, no. 3, pp. 509–518, 2009.

[25] Xilinx, "UG534 - ML605 Hardware User Guide," vol. 534, 2012.

[26] E2V, "Quad ADC - EV8AQ160 Datasheet," 2010.

[27] W. Hayward and D. Bingham, "Direct Conversion, A Neglected Technique," *QST, November*, 1968.

[28] T. Meng, "Direct-conversion RF receiver design," *IEEE Transactions on Communications*, vol. 49, pp. 518–529, Mar. 2001.

[29] D. Goodman and M. Carey, "Nine digital filters for decimation and interpolation," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 25, pp. 121–126, Apr. 1977.

[30] R. Vaughan, N. Scott, and D. White, "The theory of bandpass sampling," *IEEE Transactions on Signal Processing*, vol. 39, no. 9, pp. 1973–1984, 1991.

[31] D. Akos and M. Stockmaster, "Direct bandpass sampling of multiple distinct RF signals," *IEEE Transactions on Communications*, vol. 47, pp. 983–988, July 1999.

[32] M. Mishali, Y. Eldar, O. Dounaevsky, and E. Shoshan, "Xampling: Analog to digital at sub-Nyquist rates," *IET circuits, devices & systems*, no. 751, 2011.

[33] G. Hill, "The benefits of undersampling," *Electronic Design*, vol. 42, no. 14, 1994.

[34] Abilis, "Abilis Systems and MaxLinear introduce a reference platform for satellite home gateways that distributes up to eight satellite TV channels to IP-connected devices," 2013.

[35] T. Bijlsma, P. Wolkotte, and G. Smit, "An optimal architecture for a DDC," *Parallel and Distributed Processing Symposium*, pp. 6–13, 2006.

[36] J. Chen, D. Wu, G. Chen, Z. Jin, and X. Liu, "A 6-GHz 64×3-bit ROM for DDS application in GaAs technology," *International Conference on Microwave and Millimeter Wave Technology*, pp. 539–542, May 2010.

[37] Xilinx, "DS246 - DDS v5.0 Product Specification," pp. 1–26, 2005.

[38] F. Cardells-tormo, J. Valls-coquillat, and C. D. Vera, "Optimized FPGA-implementation of quadrature DDS," *IEEE International Symposium on Circuits and Systems*, no. 1, pp. 369–372, 2002.

[39] S. M. Badave and A. S. Bhalchandra, "Multiplierless FIR Filter Implementation on FPGA," vol. 2, no. 3, pp. 456–459, 2012.

[40] E. Garcia, R. Cumplido, and M. Arias, "Pipelined CORDIC design on FPGA for a digital sine and cosine waves generator," *3rd International Conference on Electrical and Electronics Engineering*, vol. 1, pp. 1–4, Sept. 2006.

[41] Various, "Wikipedia," 2001.

[42] A. Rudra, "Multichannel, multiband VHF software-radio-based receiver eliminates RF down conversion," *RF DESIGN*, no. July, pp. 2–7, 2003.

[43] J. Halamek, I. Viscor, and M. Kasal, "Dynamic range and acquisition system," *Measurement Science Review*, vol. 1, no. 1, pp. 71–74, 2001.

[44] W. Kester, "MT-001: Taking the Mystery out of the Infamous Formula," SNR= 6.02 N+ 1.76 dB," and Why You Should Care," 2005.

[45] W. Kester, "MT-006: ADC Noise Figure - An Often Misunderstood and Misinterpreted Specification," pp. 1–9, 2008.

[46] W. Kester, "MT-010: The Importance of Data Converter Static Specifications - Don't Lose Sight of the Basics!," pp. 1–9, 2005.

[47] W. G. Measurement, A. Technical, and M. Society, *IEEE Standard for Terminology and Test Methods for Analog-to-Digital IEEE Instrumentation & Measurement Society.* No. January, 2011.

[48] J. Fisk, "Receiver Noise Figure Sensitivity and Dynamic Range-What the Numbers Mean," *Ham Radio, October*, no. October 1975, pp. 1–18, 1975.

[49] W. Kester, "Intermodulation Distortion Considerations for ADCs MT-012," pp. 10–15.

[50] S. S. M. F. Dia, C. Thibeault, and F. F. Gagnon, "A Very High Speed and Efficient CIC Decimation Filter Core," in *IEEE North-East Workshop on Circuits and Systems*, pp. 61–64, June 2006.

[51] S. Shanthala and S. Kulkarni, "High speed and low power FPGA implementation of FIR filter for DSP applications," *European Journal of Scientific Research*, vol. 31, no. 1, pp. 19–28, 2009.

[52] M. Oberweger, A. Buchegger, and N. Heran, "Comparison of Filter Structures for FIR Filters by using a Cascade-Form FIR Filter," pp. 0–4, 2011.

[53] Xilinx, "UG363 - Virtex-6 FPGA Memory Resources," vol. 363, pp. 1–84, 2013.

[54] M. Löhning, T. Hentschel, and G. Fettweis, "Digital down conversion in software radio terminals," *Proceedings of the 10. European Signal Processing Conference (EUSIPCO)*, vol. 0, no. 5, pp. 7–10, 2000.

[55] Xilinx, "DS858 - LogiCORE IP CORDIC v5.0," 2011.

[56] 4DSP, "4-channel 8-bit ADC - 5 Gsps Product Overview."

[57] Microwaves, "Microwaves101," *http://www.microwaves101.com/encyclopedia/ampdirectivity.cfm.*

[58] Microchip, "2K I2C Serial EEPROM, 24AA02/24LC02B Datasheet," pp. 1–24, 2003.

[59] AD, "Temperature Sensor and 8-Channel ADC - ADT7411 Datasheet," 2006.

[60] NXP, "I2C-bus to SPI bridge - SC18IS602B Datasheet," 2010.

[61] AnalogDevices, "12-Output Clock Generator with Integrated 2 . 8 GHz VCO - AD9517 Datasheet," 2013.

[62] Xilinx, "UG535 - ML605 Reference Design User Guide," 2009.

[63] Ettus, "USRP1," *https://www.ettus.com/product/details/USRPPKG*.

[64] Rohde&Schwarz, "R&S SMBV100A - Vector Signal Generator Specifications," 2013.

[65] K. Lundberg, "High-speed analog-to-digital converter survey," vol. 2, pp. 1–10, 2006.

[66] T. Neu, "Clock jitter analyzed in the time domain," tech. rep., 2010.

[67] "GNU-Radio Companion."

[68] W. Kester, "MT-003: understand SINAD, ENOB, SNR, THD, THD+ N, and SFDR, so you don't get lost in the noise floor," *http.//www.analog.com/en/content/0*, 2005.

[69] F. Harris, "On the use of windows for harmonic analysis with the discrete Fourier transform," *Proceedings of the IEEE*, 1978.

[70] LDS-Group, "Understanding FFT Windows," 2003.

[71] W. Kester, "MT-007: Aperture Time, Aperture Jitter, Aperture Delay Time - Removing the Confusion," pp. 1–8.

[72] V. Arkesteijn, *Analog front-ends for software-defined radio receivers*. PhD thesis, 2007.

[73] B. Setterberg, "A 14b 2.5 GS/s 8-way-interleaved pipelined ADC with background calibration and digital dynamic linearity correction," *IEEE International Solid-State Circuits Conference Digest of Technical Papers (ISSCC)*, pp. 466–468, 2013.

[74] J. Wu, A. Chou, C. Yang, and Y. Ding, "A 5.4 GS/s 12b 500mW pipeline ADC in 28nm CMOS," *Symposium on VLSI Circuits (VLSIC)*, pp. 92–93, 2013.

[75] A. Abumurad and K. Choi, "Increasing the ADC precision with oversampling in a flash ADC," *IEEE 11th International Conference on Solid-State and Integrated Circuit Technology*, pp. 1–4, Oct. 2012.