Examensarbete utfört i Elektroteknik
vid Tekniska högskolan vid Linköpings universitet
av
Johan Nilsson och Mikael Rothin

LiTH-ISY-EX--12/4570--SE
Linköping 2012
Live Demonstration of Mismatch Compensation for Time-Interleaved ADCs

Examensarbete utfört i Elektroteknik vid Tekniska högskolan i Linköping
av

Johan Nilsson och Mikael Rothin

LiTH-ISY-EX--12/4570--SE

Handledare: Amir Eghbali
isy, Linköpings universitet
Frida Eng
Signal Processing Devices Sweden AB

Examinator: Håkan Johansson
isy, Linköpings universitet

Linköping, 7 June, 2012
The purpose of this thesis is to demonstrate the effects of mismatch errors that occur in time-interleaved analog-to-digital converters (TI-ADC) and how these are compensated for by proprietary methods from Signal Processing Devices Sweden AB. This will be demonstrated by two different implementations, both based on the combined digitizer/generator SDR14. These demonstrations shall be done in a way that is easy to grasp for people with limited knowledge in signal processing. The first implementation is an analog video demo where an analog video signal is sampled by such an TI-ADC in the SDR14, and then converted back to analog and displayed with the help of a TV tuner. The mismatch compensation can be turned on and off and the difference on the resulting video image is clearly visible.

The second implementation is a digital communication demo based on W-CDMA, implemented on the FPGA of the SDR14. Four parallel W-CDMA signals of 5 MHz are sent and received by the SDR14. QPSK, 16-QAM, and 64-QAM modulated signals were successfully sent and the mismatch effects were clearly visible in the constellation diagrams. Techniques used are, for example: root-raised cosine pulse shaping, RF modulation, carrier recovery, and timing recovery.
Abstract

The purpose of this thesis is to demonstrate the effects of mismatch errors that occur in time-interleaved analog-to-digital converters (TI-ADC) and how these are compensated for by proprietary methods from Signal Processing Devices Sweden AB. This will be demonstrated by two different implementations, both based on the combined digitizer/generator SDR14. These demonstrations shall be done in a way that is easy to grasp for people with limited knowledge in signal processing.

The first implementation is an analog video demo where an analog video signal is sampled by such an TI-ADC in the SDR14, and then converted back to analog and displayed with the help of a TV tuner. The mismatch compensation can be turned on and off and the difference on the resulting video image is clearly visible.

The second implementation is a digital communication demo based on W-CDMA, implemented on the FPGA of the SDR14. Four parallel W-CDMA signals of 5 MHz are sent and received by the SDR14. QPSK, 16-QAM, and 64-QAM modulated signals were successfully sent and the mismatch effects were clearly visible in the constellation diagrams. Techniques used are, for example: root-raised cosine pulse shaping, RF modulation, carrier recovery, and timing recovery.
Acknowledgments

First we want to thank Frida Eng and Anders Björklid at SP Devices for great help and support, especially with the report. We would also like to thank our supervisor Amir Eghbali and examiner Håkan Johansson at the Division of Electronics Systems at Linköping University. Great thanks goes out to SP Devices and all of the personnel for giving us this opportunity and for help and support. A special thanks to Daniel Björklund for inspiring discussions and good collaboration during our theses.

At last, and the most important, we extend our heartfelt thanks to our families and friends.
# Abbreviations

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>3GPP</td>
<td>3rd Generation Partnership Project</td>
</tr>
<tr>
<td>ADC</td>
<td>Analog-to-Digital Converter</td>
</tr>
<tr>
<td>API</td>
<td>Application Programming Interface</td>
</tr>
<tr>
<td>AWGN</td>
<td>Additive White Gaussian Noise</td>
</tr>
<tr>
<td>BB</td>
<td>Baseband</td>
</tr>
<tr>
<td>BER</td>
<td>Bit Error Rate</td>
</tr>
<tr>
<td>BPSK</td>
<td>Binary Phase Shift Keying</td>
</tr>
<tr>
<td>CDMA</td>
<td>Code Division Multiple Access</td>
</tr>
<tr>
<td>DAC</td>
<td>Digital-to-Analog Converter</td>
</tr>
<tr>
<td>DC</td>
<td>Direct Current</td>
</tr>
<tr>
<td>DD</td>
<td>Decision-Directed</td>
</tr>
<tr>
<td>DDC</td>
<td>Digital Down Converter</td>
</tr>
<tr>
<td>DDS</td>
<td>Direct Digital Synthesizer</td>
</tr>
<tr>
<td>DPSK</td>
<td>Differential Phase-Shift Keying</td>
</tr>
<tr>
<td>DSP</td>
<td>Digital Signal Processing</td>
</tr>
<tr>
<td>DUC</td>
<td>Digital Up Converter</td>
</tr>
<tr>
<td>FDMA</td>
<td>Frequency Division Multiple Access</td>
</tr>
<tr>
<td>FFT</td>
<td>Fast Fourier Transform</td>
</tr>
<tr>
<td>FIR</td>
<td>Finite-length Impulse Response</td>
</tr>
<tr>
<td>FPGA</td>
<td>Field-Programmable Gate Array</td>
</tr>
<tr>
<td>GPIO</td>
<td>General Purpose Input Output</td>
</tr>
<tr>
<td>GSPS</td>
<td>Giga Samples Per Second</td>
</tr>
<tr>
<td>HDL</td>
<td>Hardware Description Language</td>
</tr>
<tr>
<td>HSPA</td>
<td>High Speed Packet Access</td>
</tr>
<tr>
<td>IP</td>
<td>Intellectual Property</td>
</tr>
<tr>
<td>LFSR</td>
<td>Linear Feedback Shift Register</td>
</tr>
<tr>
<td>LP</td>
<td>Low Pass</td>
</tr>
<tr>
<td>MSPS</td>
<td>Mega Samples Per Second</td>
</tr>
<tr>
<td>NDD</td>
<td>Non-Decision-Directed</td>
</tr>
<tr>
<td>NTSC</td>
<td>National Television System Committee</td>
</tr>
<tr>
<td>PAL</td>
<td>Phase Alternating Line</td>
</tr>
<tr>
<td>PLL</td>
<td>Phase-Looked Loop</td>
</tr>
<tr>
<td>PRBS</td>
<td>Pseudo Random Binary Sequence</td>
</tr>
</tbody>
</table>
Abbreviations

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>PSK</td>
<td>Phase-Shift Keying</td>
</tr>
<tr>
<td>QAM</td>
<td>Quadrature Amplitude Modulation</td>
</tr>
<tr>
<td>QPSK</td>
<td>Quadrature Phase-Shift Keying</td>
</tr>
<tr>
<td>QQAM</td>
<td>Quotient Quadrature Phase-Shift Keying</td>
</tr>
<tr>
<td>RC</td>
<td>Raised Cosine</td>
</tr>
<tr>
<td>RF</td>
<td>Radio Frequency</td>
</tr>
<tr>
<td>RRC</td>
<td>Root-Raised Cosine</td>
</tr>
<tr>
<td>SDR</td>
<td>Software Defined Radio</td>
</tr>
<tr>
<td>SECAM</td>
<td>Sequential Color with Memory/Séquentiel Couleur À Mémoire</td>
</tr>
<tr>
<td>SNR</td>
<td>Signal-to-Noise Ratio</td>
</tr>
<tr>
<td>TDMA</td>
<td>Time Division Multiple Access</td>
</tr>
<tr>
<td>TI-ADC</td>
<td>Time-Interleaved Analog-to-Digital converter</td>
</tr>
<tr>
<td>W-CDMA</td>
<td>Wideband Code Division Multiple Access</td>
</tr>
</tbody>
</table>
## Contents

1 Introduction  
  1.1 Background ........................................ 1  
  1.2 Purpose ............................................ 2  
  1.3 Requirements ....................................... 2  
    1.3.1 Analog Video Demo ............................ 3  
    1.3.2 Digital Communication Demo .................. 3  
  1.4 Tools ............................................... 3  
    1.4.1 Software Tools .................................. 3  
    1.4.2 Hardware Tools ................................ 5  
  1.5 Disposition ....................................... 5  

2 Time-Interleaved ADCs  
  2.1 Motivation ......................................... 7  
  2.2 The Channel Mismatch Problem ...................... 8  
    2.2.1 Offset Error .................................... 8  
    2.2.2 Time Skew Error ............................... 8  
    2.2.3 Gain Error ..................................... 9  
    2.2.4 Bandwidth Mismatch ........................... 9  
  2.3 Solution - ADX ...................................... 9  
  2.4 SDR14 ................................................ 10  

3 Analog Video Demo  
  3.1 Analog Video ....................................... 13  
    3.1.1 Different Standards ............................ 13  
    3.1.2 RF Modulation of Composite Video .......... 14  
    3.1.3 Decisions ...................................... 14  
  3.2 Implementation ..................................... 15  
    3.2.1 Motivation ..................................... 15  
    3.2.2 Implementation ................................ 16  
  3.3 Results ............................................. 17  

4 Digital Communication Systems  
  4.1 Source and Source Coding ......................... 20  
  4.2 Channel Coding .................................... 20  
  4.3 Mapping ............................................. 20
## Contents

4.3.1 PSK ................................................. 20  
4.3.2 QAM .................................................. 21  
4.4 Pulse Shaping ........................................... 22  
4.4.1 Raised-Cosine Filter ................................. 23  
4.4.2 Matched Filtering .................................... 24  
4.4.3 Inter-Symbol Interference ............................ 25  
4.5 RF Modulation .......................................... 26  
4.6 Channel ................................................. 27  
4.7 Multiple Access Techniques .............................. 27  
4.7.1 TDMA ................................................. 28  
4.7.2 FDMA .................................................. 28  
4.7.3 CDMA .................................................. 28  
4.8 Synchronization .......................................... 29  
4.8.1 Carrier Recovery ...................................... 29  
4.8.2 Timing Recovery ...................................... 33  
5 Digital Communication Demo - Implementation .......... 37  
5.1 Motivation and Decisions ................................. 37  
5.1.1 Source and Source Coding ............................. 38  
5.1.2 Channel Coding ....................................... 38  
5.1.3 Mapping ............................................... 39  
5.1.4 Multiple Access Techniques .......................... 39  
5.1.5 Pulse Shaping ......................................... 39  
5.1.6 RF Modulation ........................................ 40  
5.1.7 Physical Channel ...................................... 40  
5.1.8 Synchronization ....................................... 40  
5.2 Hardware ................................................. 41  
5.2.1 The FPGA of the SDR14 ............................ 41  
5.2.2 The DSP48E1 Block .................................... 42  
5.3 Implementation/Design .................................. 43  
5.3.1 PC Interface ......................................... 45  
5.3.2 Symbol Generation .................................... 47  
5.3.3 Sample Rate Conversion and Filtering .............. 47  
5.3.4 Mapping Filters to Hardware ........................ 51  
5.3.5 Baseband Mixer and DDS ............................. 53  
5.3.6 Timing Recovery ..................................... 54  
5.3.7 Carrier Recovery ..................................... 55  
5.3.8 Word Lengths ........................................ 56  
6 Digital Communication Demo - Tests and Results ......... 57  
6.1 ADX IP Demo ........................................... 57  
6.2 Digital Transmitter and Receiver ........................ 61  
6.2.1 AWGN in Simulink .................................. 62  
6.2.2 Inter-Channel Interference .......................... 62  
6.2.3 W-CDMA Compliance ................................. 64  
6.3 Carrier Recovery ....................................... 64
 Contents                                                                 xiii  

  6.4 Timing Recovery .................................................. 65  
  6.5 Resource Utilization ......................................... 66  

  7 Conclusions  .................................................... 67  
    7.1 Analog Video Demo ........................................... 67  
    7.2 Digital Communication Demo  .......................... 68  
      7.2.1 W-CDMA .................................................. 68  
      7.2.2 Bandwidth ............................................... 68  
      7.2.3 Carrier Recovery ...................................... 68  
      7.2.4 Timing Recovery ...................................... 69  
    7.3 Future Work .................................................. 69  

 Bibliography ....................................................... 71
**List of Figures**

1.1 FFT of the received signal, sinusoidal at 345 MHz. .......................... 2
1.2 Screen shot from ADCaptureLab. ............................................. 4

2.1 Time-interleaved ADC. ............................................................... 7
2.2 Offset error in a time-interleaved ADC. ....................................... 8
2.3 Time skew error in a time-interleaved ADC. ................................. 9
2.4 Gain error in a time-interleaved ADC. ....................................... 9
2.5 ADX overview. ........................................................................ 10
2.6 Photo of SDR14. ................................................................. 10

3.1 PAL-B spectrum. .................................................................... 14
3.2 Video signal and sinusoid. .......................................................... 15
3.3 Analog video demo setup schematic. ...................................... 16
3.4 Photo of the analog video demo setup. ........................................ 17
3.5 Analog result with a sinusoid at 341.7 MHz and ADX IP deactivated. 18
3.6 Analog result with a sinusoid at 341.7 MHz and ADX IP activated. 18

4.1 Overview of a typical digital communication system. ................. 19
4.2 Blocks within the modulation block. ........................................... 19
4.3 Grey-coded QPSK constellation diagram. .................................. 21
4.4 Grey-coded 16-QAM constellation diagram. ............................. 21
4.5 I part of a QPSK modulated signal. .......................................... 22
4.6 Pulse shaped signal .................................................................. 22
4.7 Frequency responses of RRC filter for different $\alpha$. ............... 24
4.8 Demonstration of inter-symbol interference properties of the RC filter. 25
4.9 Complex-valued modulation. .................................................. 27
4.10 Magnitude response of $x_{rf}(t)$. ............................................ 27
4.11 Phase offset at receiver mixer with QPSK signal. ....................... 29
4.12 Frequency offset at receiver mixer with QPSK signal. .............. 30
4.13 Basic PLL. ........................................................................... 31
4.14 Costas loop for BPSK carrier tracking. .................................... 31
4.15 Spilkers loop for QPSK carrier tracking. ................................. 31
4.16 Phase offset at receiver mixer with QAM signal. ....................... 32
4.17 Multiply-filter-divide circuitry for M-ary PSK. .......................... 33
4.18 Early-late gate algorithm for the ideal, the early and the late case. 34
4.19 The Mueller and Muller algorithm for the ideal, the slow and the fast case. ................................................................. 34
4.20 Gardner algorithm for the ideal, the early and the late case. ....... 35

5.1 Frequency spectrum of the received signal. Solid lines: Spectrum before ADCs. Solid + Dashed lines: Spectrum after the ADCs. ... 38
5.2 SDR14 block diagram. .......................................................... 42
5.3 Transmitter. ................................................................. 43
5.4 Receiver. ........................................................................... 43
5.5 Baseband modulator. .......................................................... 44
Contents

5.6 Baseband demodulator. 44
5.7 Frequency spectrum of four channels from four baseband modulator blocks, added together and then modulated by the DUC to 130 MHz. 45
5.8 The LFSR used for symbol generation. 47
5.9 Magnitude response of the RRC filter implemented in hardware. 48
5.10 Frequency spectrum of four channels (A, B, C, and D) demonstrating aliasing when decimating by 13 after mixing with 7.5 MHz. 49
5.11 Magnitude response of the LP$_{13}$ filter implemented in hardware. 50
5.12 Magnitude response of the LP$_{8}$ filter implemented in hardware. 51
5.13 Impulse response of the LP$_{8}$ filter. 52
5.14 Direct-form FIR filter. 53
5.15 Symmetric FIR filter with reduced number of multipliers. 53
5.16 Baseband mixer implementation. 54

6.1 Simulated interleaving distortion in Simulink. 58
6.2 Test setup. 58
6.3 Signal from the TI-ADC, ADX is deactivated. 59
6.4 Constellation diagrams for the two channels in Fig. 6.3 with ADX deactivated. 59
6.5 Constellation diagrams for the two channels in Fig. 6.3 with ADX activated. 60
6.6 Test of the available mappings. 61
6.7 AWGN test in Simulink with an SNR of 0 dB and 3 dB. 62
6.8 FFT plot showing inter-channel interference. 63
6.9 Test of inter-channel interference, constellation diagrams. 63
6.10 Signal from W-CDMA signal generator with frequency error. Carrier recovery active or not. 65
6.11 Simulink simulation of different timing offset values. 66
Chapter 1

Introduction

Analog-to-digital converters (ADCs) are important components in modern signal processing. An effective way of increasing the performance in terms of conversions per second is to interleave several converters in time. This is called a time-interleaved analog-to-digital converter (TI-ADC). As a downside, distortion is introduced due to component mismatches among the time-interleaved converters. This degrades the performance in terms of resolution. To avoid this and maintain the resolution, the errors must be compensated for in some way.

This masters thesis is carried out at Signal Processing Devices Sweden AB, from here on referred to as SP Devices, who are specialized in algorithms for this mismatch compensation. This makes them able to develop products with cutting edge sampling rates at high resolutions. The purpose of this thesis is to visualize the difference between using or not using their algorithms, by implementing demonstration applications on a product from the company.

1.1 Background

SP Devices was founded in 2004 by four people from Linköping University. Their product was an algorithm to solve the problems introduced when using TI-ADCs and was based on research carried out at the university. To this date, SP Devices has grown to about 20 employees and makes business with several large companies in the field. The algorithm is still in focus and is sold either as an intellectual property (IP) block for field programmable gate arrays (FPGAs), silicon or software, as part of their digitizer products, or in a system designed on request.

One of SP Devices newest products, SDR14, is a combined digitizer and generator which basically means a unit with the capability of both converting analog input signals to digital data, and to output analog signals from digital data. It has two analog inputs and two analog outputs. The TI-ADCs have a sample rate of 800 MSPS and the digital-to-analog converters (DACs) has a sample rate of 1600 MSPS. It also contains a customizable FPGA. This makes it possible to not only collect data, but also to pass a signal through for processing or use it as a digital transceiver. Both are done in this thesis.
1.2 Purpose

It is important to be able to demonstrate how well the algorithms perform. This can, for example, be done by using spectrum plots like the ones in Fig. 1.1. But for people with limited knowledge in signal processing it can be hard to grasp what this really means for a certain application. Therefore, an implementation for demonstration is requested which transfers some signal that can be visualized and is distorted due to the mismatch effects.

![Figure 1.1: FFT of the received signal, sinusoidal at 345 MHz.](image)

(a) Unwanted frequency component at 55 MHz due to the TI-ADC.  
(b) Algorithm from SP Devices active. Unwanted frequency component removed.

Two implementations are chosen. The first implementation is based on an analog video signal and the second implementation, which is the main task in this thesis, consists of parts of a digital communications system. This second task will both demonstrate how well the algorithm performs and how suitable the digitizer/generator unit is for digital transmitter and/or receiver purposes. This means that the performance of the implemented system should be evaluated as well.

1.3 Requirements

The main requirement for this thesis is that it shall make use of the SDR14 device at SP Devices, and in a clear way demonstrate how the algorithms in this improve the operation. As described in Section 1.2, two implementations are chosen. Requirements for each of those two demo implementations are described under the following two sections.
1.4 Tools

1.3.1 Analog Video Demo

- The transmitted signal shall make use of a commonly used standard. This will make it easier to find existing products and also easier to relate to real-world applications.

1.3.2 Digital Communication Demo

- It shall demonstrate how the points in a constellation diagram are affected by spurious frequency components described in Chapter 2. This should be demonstrated live which means that all signal processing is performed in real-time.
- The transmitted analog signal shall have the frequency properties of a known and common communication standard to be able to relate to real-world applications.
- The signal shall be of as high bandwidth as possible to be a more useful signal generator. Thus, if there are enough resources available on the FPGA, several channels can be sent in parallel to increase the bandwidth.

1.4 Tools

Several tools have been used in this thesis work. How the most important and interesting ones have been used will be described here.

1.4.1 Software Tools

ADCaptureLab

ADCaptureLab is a data capture software provided together with digitizers from SP Devices. The software allows a PC, with a connected digitizer, to act as an oscilloscope and a spectrum analyzer. It has a graphical user interface as shown in Fig. 1.2. It is possible to start and stop the capture, zoom in and out on the plots and adjust how to calculate the Fast Fourier Transform (FFT). It is also possible to adjust settings of the digitizer. For example, one can choose to bypass the ADX IP block of the digitizer which is where the mismatch effects are compensated for.
Matlab and Simulink

Almost all functionality in the resulting digital demo implementation has been developed in three steps: Matlab, Simulink model, and hardware description language (HDL) code. Algorithms and filters have been implemented and tested individually in Matlab. Then the whole system has been modeled in Simulink. Finally, this implementation has been written in HDL code and simulated before applied in hardware. The results have been compared between these steps to verify the correct functionality.

Xilinx ISE

Xilinx ISE is a development environment for FPGAs. The environment contains tools for translating, synthesizing and routing HDL code onto Xilinx FPGAs. Xilinx ISE includes a tool, CORE Generator, which is used to generate configurable blocks. In this thesis CORE Generator was used to, for example, generate finite-length impulse response (FIR) filters, complex multipliers, and customized RAMs. These units are described further in Chapter 5.
ISim

ISim is a HDL simulator from Xilinx which allows functional and timing simulations. The HDL code that is written is based on the developed Simulink model. To verify that the behavior of the HDL code was the same as the Simulink model, the HDL code was simulated both in terms of function and timing. By exporting samples from one of Simulink or ISim and importing it to the other, simulations could also be performed with parts of the system in HDL code and parts of the system in a Simulink model to verify the behavior of some modules.

1.4.2 Hardware Tools

Signal generators

To generate the signal that shall introduce mismatch effects on the transmitted signal, a signal generator was used which generates a sinusoidal signal with an amplitude of up to 1 V and a frequency up to 1.2 GHz. To verify the implemented receiver, a signal generator capable of generating a signal similar to a wideband code division multiple access (W-CDMA) signal was used.

Oscilloscope and Spectrum scope

These tools have been very useful in order to verify that the generated signals are correct. Even though the digitizer from SP Devices should perform this as well, it is using a customizable FPGA where the rest of the system is also implemented. This makes it possible to interfere with the signals in a way that could give false results.

1.5 Disposition

Chapter 2 - Time-Interleaved ADCs introduces the TI-ADC and the problems related to it. The algorithm from SP Devices to correct for these problems is also presented.

Chapter 3 - Analog Video Demo describes the implementation and the results from the demo implementation with analog video.

Chapter 4 - Digital Communication Systems covers the basic theory behind a digital communication system. Typical components are described and possible solutions to some known problems are introduced.

Chapter 5 - Digital communication Demo - Implementation describes the implementation of the second demo implementation, based on a digital communication system. Some implementation details of the solutions described in Chapter 4 are presented.

Chapter 6 - Digital communication Demo - Tests and Results shows the results from tests of the system described in Chapter 5.

Chapter 7 - Conclusions contains a summary of the results. There is also a discussion about these results and some possible future work.
Chapter 2

Time-Interleaved ADCs

This chapter presents the method to interleave several ADCs in time to improve the performance of analog-to-digital conversion. Problems with this method are discussed and the solution by SP Devices is presented.

2.1 Motivation

One important performance metric of analog-to-digital converters is the sample rate. The reason for this is, for example, that if one would like to perfectly convert a bandlimited signal with a highest frequency component of \( f \), one need a sample rate larger than \( 2f \). This is according to the Nyquist-Shannon sampling theorem. So to correctly receive signals with high frequency components you need a high sample rate. One way of increasing the sample rate at a low cost is to interleave \( M \) ADCs in time [1], as shown in Fig. 2.1. This means that the sample rate is increased by \( M \) times. This solution can also be cheaper in terms of power and area usage, compared to the case of a single ADC operating at \( M \) times the rate of each of the interleaved ADCs.

![Figure 2.1: Time-interleaved ADC.](image-url)
2.2 The Channel Mismatch Problem

When using several ADCs in parallel, some specific errors, which are not present in the case of one single ADC, are introduced due to component differences. These errors are called mismatch errors and affect the performance of the time interleaved ADC by introducing additional spurious frequency components (spurs) which affect the resolution. There are three main mismatch errors: time skew, gain and offset errors. These are described in Sections 2.2.1 - 2.2.3. A fourth error originating from bandwidth mismatch in the sample and hold circuitry is described in Section 2.2.4. In the equations below, $f_{\text{dist}}$ is the frequency where the frequency component $f_{\text{in}}$ will generate a spur.

2.2.1 Offset Error

The offset error originates from the fact that each ADC has a small offset from DC. This offset varies between different ADCs. The offset error is illustrated in Fig. 2.2. This error results in distortion which in the frequency domain corresponds to spurs at frequencies

$$f_{\text{dist}} = k\frac{f_s}{M}, \quad k = 1, 2, \ldots, M - 1.$$  \hspace{1cm} (2.1)

As seen in (2.1), this is independent of the input signal. [2]

Figure 2.2: Offset error in a time-interleaved ADC.

2.2.2 Time Skew Error

Time skew means that the time between samples, taken by the time-interleaved ADC, is not constant, even though the time between samples, for each ADC, is constant (clock jitter ignored). This is shown in Fig. 2.3. This error results in distortion with the highest amplitude at zero-crossings in the time domain. In the frequency domain, this results in spurs centered at

$$f_{\text{dist}} = \pm f_{\text{in}} + k\frac{f_s}{M}, \quad k = 1, 2, \ldots, M - 1.$$  \hspace{1cm} (2.2)

Both the size and location of the spurs are dependent on the input signal frequency. A signal with a high frequency will generate a higher error compared to a signal with a low frequency. [2]
2.2.3 Gain Error

The gain error occurs in the case of different gains in the different ADCs. This means that a certain voltage does not result in the same digital value for different ADCs. An example of this error is illustrated in Fig. 2.4. This error results in distortion with the highest amplitude at the wave peaks in the time domain. In the frequency domain, this results in spurs centered at the same frequencies as the time skew error, see (2.2). [2]

2.2.4 Bandwidth Mismatch

Another error, which does not really originate from the ADCs is the bandwidth mismatch error. This is instead due to differences in the sample-and-hold circuits before the ADCs. This generates spurs at the same frequencies as the time skew and gain offset errors, see (2.2). [2]

2.3 Solution - ADX

To be able to use time-interleaved ADCs with a high precision, the mismatches described in this chapter must be compensated for. SP Devices has a patent on a method compensating for this [3]. The method is basically to see the errors introduced as $M$ filters applied to the input signal periodically over time (different for each ADC), and estimate these filters. Then, filters which are the inverse of
these estimated filters are used to reconstruct the signal by removing these errors. A simple overview of a system interleaving four ADCs in time is shown in Fig. 2.5. The estimator analyzes the input signal and generates filter coefficients for the reconstructor. This is done all digital which means that it can be done in software or as an IP block on FPGA or silicon.

![Figure 2.5: ADX overview.](image1)

![Figure 2.6: Photo of SDR14.](image2)

### 2.4 SDR14

Together with IP blocks for removing the distortions related to interleaving of ADCs, SP Devices also develops digitizers focused on high speed and high resolution. SDR14 is such a device which has two analog inputs and two analog outputs. For a photo of the device, see Fig. 2.6. To achieve a high sample rate, each channel...
has two time-interleaved ADCs. According to the data sheet [4], this configuration gives the device an input sample rate of 800 MSPS. The analog input bandwidth is 600 MHz and without undersampling it is possible to sample and reconstruct signals with frequencies up to 400 MHz, according to the Nyquist-Shannon sampling theorem. The analog output bandwidth is 400 MHz and the update rate of the DACs is 1.6 GSPS. The device is equipped with a Virtex-6 LX240T FPGA from Xilinx, which allows customized digital signal processing of the received data. In this thesis the SDR14 device is, for example, used as a transmitter and a receiver for a digitally modulated signal, as described later in Chapter 5.
Chapter 3

Analog Video Demo

To demonstrate the distortion effects on a signal described in Chapter 2, a choice was made to use a video signal. It should be safe to say that most people know the difference between a good and a bad video signal when looking at the image, which makes it good for demonstration purposes. However, the modern digital video signals often include error correction and is modulated in quite advanced ways. An example is digital video broadcasting-terrestrial (DVB-T), which is described in [5]. Therefore, it should be more practical to use an older analog video signal without this to make it simpler to derive the effects on the image from the size and frequency of the distortions.

This chapter introduces analog video standards, the demo implementation, and finally the results of the demo.

3.1 Analog Video

In this section, analog video standards are discussed. It is also discussed how these are transmitted at higher frequencies and which decisions to make for this demo implementation in terms of standards and frequencies.

3.1.1 Different Standards

When transmitting analog video, many different kinds of cables can be used depending on the standard. Most of the standards rely on several pins to separate red, green, and blue colors or luminance and chrominance (or other combinations). For the analog video demo, the desired standard should be able to transmit on a single cable. The only suitable standard found for this was composite video.

Composite video is the video signal widely used in analog television, described further in [6]. It consists of luminance and chrominance signals. The luminance, or brightness, is single sided with a bandwidth of 5 MHz. The chrominance is a combination of two color signals (The difference between red and the luminance and the difference between blue and the luminance) and has a bandwidth of 1 MHz. These two are modulated by quadrature amplitude modulation (QAM), see
Section 4.3.2, to a single carrier at 4.43 MHz above the luminance carrier. This results in a composite color video signal with a total bandwidth of 6 MHz. In the PAL standard, composite video is combined with an audio channel located 1 MHz above the chrominance signal. Figure 3.1 shows the spectrum of the complete PAL signal. There exists several different PAL standards but PAL-B is used in this thesis.

Figure 3.1: PAL-B spectrum.

3.1.2 RF Modulation of Composite Video

To be able to transmit a signal through the air or, as in this case, to pass through the ADCs and DACs of the SDR14, the signal needs to be modulated onto a radio frequency (RF) carrier. The composite video signal described in Section 3.1.1 is suitable to be modulated onto an RF carrier. This is because it is a single baseband signal of 6 MHz and not, for example, divided over many cables. The RF modulation of composite video is, for example, done in the PAL standard but also in other widely used television systems like NTSC and SECAM described in [7]. The wide use of these standards is good since, in this demo application, the signal should preferably be located on some RF to work with the equipment used. There is also a common problem (at least some years ago) that some televisions did only provide RF input and some video sources did not provide RF output (but composite output). This means that there exist products on the consumer market for modulating a composite video signal to RF.

3.1.3 Decisions

Based on the standards and techniques discussed in this chapter, the video standard chosen is composite video. The reason for this is that it appears to be the only suitable option for transmission of video on a single coaxial cable. This means that simple and cheap sources of video can be found to generate the video signal. The signal is modulated with an RF carrier to be able to pass through the ADCs and DACs of the SDR14 and as mentioned in Section 3.1.2, there exist cheap products on the market for this as well. The video source also provides audio output which is modulated at a frequency right above the video signal by the RF modulator. The inclusion of audio will make it possible to distort this as well as the image and the color of the image. The video signal is generated and modulated according to the PAL video standard.
3.2 Implementation

This section describes the analog demo implementation. It starts with a description of the equipment used, then the expected results, how it was carried out, and finally a discussion about the results.

3.2.1 Motivation

The advantage of using SP Devices ADX technology, described in Chapter 2, will be demonstrated with the use of a video signal, described in Section 3.1.3. For this, the SDR14 unit described in Section 2.4 will be used.

To demonstrate this, the idea is to place the video signal at the frequency $f_{\text{dist}}$ which is where spurs from a signal at frequency $f_{\text{in}}$ appear according to (2.2). The spurs will introduce distortion visible in the received video signal. In this case, the chosen commercial RF Modulator has a fixed frequency of 55.25 MHz (analog TV channel 3 in western Europe). Based on (2.2) with the SDR14 specifications $f_s = 800$ MSPS and $M = 2$, $f_{\text{in}}$ will be chosen as

$$\pm f_{\text{in}} = f_{\text{dist}} - \frac{f_s}{2} \Rightarrow$$
$$\pm f_{\text{in}} = 55.25 - 400 \Rightarrow$$
$$f_{\text{in}} = 344.75,$$  \hspace{1cm} (3.1)

where the signal at $f_{\text{in}}$ is just a pure sinusoidal signal. The selection of a regular channel used in earlier analog TV broadcast means that it should be easy to locate the video signal with any TV tuner equipment. Figure 3.2 shows how the spur that originates from the sinusoidal signal at 344.75 MHz overlaps with the video signal at 55.25 MHz.

![Figure 3.2: Video signal and sinusoid. The solid frequencies are the generated video source and the signal from the signal generator. The dashed frequencies originates from the distortion introduced in the TI-ADC.](image-url)
3.2.2 Implementation

The equipment used for the demonstration is:

- Roxcore EZ-Play Base media player (Video source)
- König AV RF Modulator
- Hameg HM8134-3 signal generator
- SP Devices SDR14 digitizer/generator
- Pinnacle PCTV 340e Hybrid Pro Stick (Receiver)
- Signal combiner

The equipment is set up as illustrated in Fig. 3.3 and a photo of the setup is shown in Fig. 3.4. The connection between the video source and RF modulator is composite video and analog stereo audio (total 3xRCA). The other connections are coaxial cables. The signal from the RF modulator and the signal generator is added together in a combiner.

![Diagram of analog video demo setup schematic]

The RF Modulator generates a video signal with an amplitude of about 50 mV peak-to-peak, the amplitude after SDR14 is about 28 mV peak-to-peak. The lower amplitude after SDR14 is because of a lower maximum output from the DAC than the maximum input of the ADC. This does not affect the image quality since the receiver compensates for different signal levels. Since the analog video signal is 6 MHz wide the signal generator is set to generate a pure sinusoid at a frequency between 338.75 and 344.75 MHz, with a peak amplitude of 0.2 V. A setting of 344.75 MHz will result in a spur at 55.25 MHz and 338.75 MHz results in a spur at 61.25 MHz.

As mentioned in Section 1.4.1, it is possible to activate and deactivate the ADX IP in the SDR14 from the PC software ADCaptureLab. This makes it possible to directly see the results of the algorithms on or off, while looking at the video image.
3.3 Results

The results of the demo implementation was very positive. It was possible to almost block the video signal completely with a signal at some of the given frequencies. Especially when the blocking signal was applied at 344.75 MHz, resulting in a spur at 55.25 MHz which is the carrier frequency of the main video signal (luminance). When applied at about 340.40 MHz instead, the effect was that color information was distorted and not the whole image. This is since the spur ends up at 59.60 MHz where the color carrier is located (chrominance). A final test was to apply the blocking signal at 338.75 MHz, thus giving a spur at 61.25 MHz and affect the audio. This was also a success and it was impossible to hear any of the original audio because of the introduced distortion. None of these effects occur when the ADX IP was active, which clearly shows the good performance. Figure 3.5 shows an example of the disturbance when the ADX IP is not active with a sinusoid located at 341.7 MHz (thus resulting in a spur at 58.3 MHz). Figure 3.6 shows the same signals but with the ADX IP activated.
Figure 3.5: Analog result with a sinusoid at 341.7 MHz and ADX IP deactivated.

Figure 3.6: Analog result with a sinusoid at 341.7 MHz and ADX IP activated.
Chapter 4

Digital Communication Systems

For the second demonstration implementation, a study of digital communication systems were done to identify which parts to implement and which existing methods to use in these parts. This chapter introduces a typical digital communication system with a transmitter and a receiver and the modules found in this, as shown in Fig. 4.1. Chapter 5 then describes the decisions made and motivates the selected components and methods. Since this thesis focuses mostly on the modulation, this is expanded into three separate blocks shown in Fig. 4.2. The corresponding blocks are present in the demodulation block as well. Each block in Fig. 4.1 and 4.2 is described in the sections below and the parts relevant for this thesis are described more thoroughly. The concepts of multiple access and synchronization techniques are also described.

![Figure 4.1: Overview of a typical digital communication system.](image)

![Figure 4.2: Blocks within the modulation block.](image)
4.1 Source and Source Coding

A source can be almost anything and the source coding is the process of converting the source data to data that is suitable for transmission through the communication system. Source coding can, for example, include compression of an input data stream to reduce the amount of data to be sent. Or it could mean something entirely different like conversion of speech to a digital data stream.

4.2 Channel Coding

When transmitting data, there is typically several different sources of noise on the channel. This noise distorts the transmitted data and introduces errors. It is often very hard and/or expensive to eliminate this, which means that the errors need to be taken care of in another way. This is called channel coding and can be done by sending extra data bits that the receiver can use to detect if the data is erroneous. To take care of incorrect data, two options exist. Either a new transmission is requested or there is enough of extra data bits sent to correct the error.

When using extra data bits for error correction, two main types of codes exist. Block codes transform a block of \( m \) bits to a block of \( n \) bits \( (n > m) \). The other type of error correction is the convolutional encoding where \( m \) input bits in a continuous data stream are mapped to \( n \) output bits \( (n > m) \) and the output bits depend on the last \( k \) bits. More about channel coding can be found in [8] and [9].

4.3 Mapping

In digital communication systems the information, typically a bit stream, which is to be transmitted has to be modulated in order to be able to send the information, for example over the air or in an analog cable. One way to do this is to map the bits to be sent to a complex signal where the real part is denoted \( I \) (In-phase) and the imaginary part \( Q \) (Quadrature-phase). The information will be modulated with one or more of these properties; the phase, the amplitude, or the frequency of these signals. Many different techniques for this mapping exist and a few of them will be described here. The selection is based on the techniques used in W-CDMA and High Speed Packet Access (HSPA). The decision to implement W-CDMA is motivated later in Section 5.1. The information carried on the \( I \) and \( Q \) signals are called symbols and the number of bits mapped to each symbol depends on the modulation method.

4.3.1 PSK

One digital modulation technique is phase shift keying (PSK). This method only allows a few different discrete phase states with the same amplitude and frequency. The number of allowed states are usually \( 2^n \), where \( n \) is the number of bits per symbol. More bits per symbol means higher possible data rate. Binary phase shift keying (BPSK) and quadrature phase shift keying (QPSK) use, respectively, two
(n=1) and four (n=2) phase states. The phase states have to be distinguished from each other even in noisy signals. Therefore, it is preferable to have the phase states separated as much as possible. The maximum difference between two adjacent states is $360^\circ/n$. That is, for example, $180^\circ$ for BPSK. In QPSK, the binary input values are mapped to values for $I$ and $Q$ according to Fig. 4.3. These values for $I$ and $Q$ are then used as input to the RF modulation described in Section 4.5. More on PSK is given in [8].

![Figure 4.3: Greycoded QPSK constellation diagram.](image)

4.3.2 QAM

Quadrature amplitude modulation (QAM) is a two dimensional modulation technique. As described in Section 4.3.1, the bits are mapped to complex numbers. A QAM-modulated signal could be described with two amplitude modulated signals $I$ and $Q$. There exist different QAM variants like 16-QAM or 64-QAM. The differences between these modulations techniques are the number of points in the complex plane. QPSK mentioned before is basically the same as 4-QAM, because

![Figure 4.4: Greycoded 16-QAM constellation diagram.](image)
all points have the same absolute amplitude and therefore, only the phase is mod-
ulated. For higher-order QAM, like 16-QAM, the amplitude is different among
the constellation points, as shown in Fig. 4.4. Just like for PSK, binary values
are mapped to values on $I$ and $Q$ which are used as input to the RF modulation
described in Section 4.5. Figure 4.4 shows the mapping. More on QAM is given
in [8].

One interesting observation is that the mapping in Fig. 4.4 is grey coded. That
is, two adjacent points only differ at one bit position. The reason for this is to
minimize the number of erroneous bits in case a symbol is interfered before it is
received and decoded back to bits from the values of $I$ and $Q$.

4.4 Pulse Shaping

Signals generated through mapping of a binary sequence to a, for example, QPSK
modulated signal is not continuous because of the discrete amplitude levels, as
shown in Fig. 4.5. If this is the case, it is impossible to transmit the signal in a

\begin{figure}[h]
\centering
\includegraphics[width=0.5\textwidth]{qpsk_signal_part.png}
\caption{I part of a QPSK modulated signal.}
\end{figure}

\begin{figure}[h]
\centering
\includegraphics[width=0.5\textwidth]{qpsk_signal_pulse_shaping.png}
\caption{Signal from Fig. 4.5 after applying pulse shaping with a raised-cosine
filter.}
\end{figure}
4.4 Pulse Shaping

band-limited channel. This is because of the discontinuities in the signal, which will make the signal non-bandlimited. A channel is, for example, a cable or the air. One way to remove the discontinuities and therefore bandlimit the signal is to perform pulse shaping to the signal. One method for pulse shaping is to upsample the signal and then low pass filter it to limit the bandwidth. An example of the result after such a filtering of the signal in Fig. 4.5 is shown in Fig. 4.6. More on digital communication in band-limited channels is discussed in [10].

4.4.1 Raised-Cosine Filter

The filter used in the pulse shaping example above is the raised-cosine (RC) filter. This filter is commonly used in digital communication systems, partially due to the inter-symbol interference properties discussed in Section 4.4.3. The impulse response for this filter is

\[
RC(t) = \frac{\sin\left(\frac{\pi t}{T}\right) \cos\left(\frac{\pi \alpha t}{T}\right)}{\pi \frac{t}{T}} = \text{sinc}\left(\frac{\pi t}{T}\right) \cos\left(\frac{\pi \alpha t}{T}\right) \frac{1}{1 - 4\alpha \left(\frac{t}{T}\right)^2} \tag{4.1}
\]

according to [10]. This filter could be split up into two filters. The relation in the frequency domain is

\[
RC(f) = |RRC(f)|^2 \tag{4.2}
\]

where \( RC(f) \) is the frequency response of the RC filter and \( RRC(f) \) is the frequency response of the new filter. These new filters are matched filters called root-raised cosine (RRC) filters and are typically used in pairs, one on the transmitter side and one on the receiver side. Both the sample time, \( T \), and the roll-off factor, \( \alpha \), affect the bandwidth of the RC filter. As shown in Fig. 4.7 the bandwidth needed for each channel increases when the parameter \( \alpha \) increases. However, it is not possible to have a too low \( \alpha \) because of the infinite length of the filter in the time domain. According to [10] a pulse occupies, for example, 35% more bandwidth with \( \alpha = 0.35 \) compared to an ideal sinc pulse (\( \alpha = 0 \)). This means that the bandwidth of the transmitted signal is the same as the symbol rate times \( 1 + \alpha \). Therefore, \( \alpha \) and \( T \) decide how much the channels have to be separated in the frequency domain to not disturb each other.
4.4.2 Matched Filtering

As mentioned above, the two RRC filters are matched filters. Matched filters are used in a method called matched filtering. The purpose of matched filtering is to maximize the signal-to-noise ratio (SNR) in a channel with Additive White Gaussian Noise (AWGN) [10]. If the received signal

\[ r(t) = x(t) + w(t) \quad 0 \leq t \leq T \]  

(4.3)

consists of the signal of interest \( x(t) \) and added noise \( w(t) \) where \( T \) is an arbitrary time interval, then the filter that optimally detects the signal \( x(t) \) is a filter that is a delayed and time-reversed version of the pulse shape of the interesting signal \( x(t) \), according to [8]. If the RRC filter \( h(t) \) is used in the transmitter, the signal \( x(t) \) will be the convolution

\[ x(t) = s(t) * h(t) = \int_{\tau=-\infty}^{\infty} s(\tau) h(t - \tau) d\tau. \]  

(4.4)

where \( s(t) \) is the sent symbols. The received signal \( r(t) \) will then be a sum of delayed and scaled RRC pulses and the noise \( w(t) \). As [8] describes, the time-reversed RRC filter is the optimal filter to maximize the SNR. This gives in the frequency domain

\[ R(f)H(-f) = (X(f) + W(f))H(-f) = S(f)H(f)H(-f) + W(f)H(-f), \]  

(4.6)

where \( H(f)H(-f) \) is the RC filter which have the property of no inter-symbol interference, as described in Section 4.4.3. In [10], Proakis describes another optimal receiver for AWGN channels, the Correlation Receiver. The output of the correlation receiver could be obtained through matched filter by sampling the output
4.4 Pulse Shaping

at the exact time point \( t = T \), as shown in Fig. 4.8. Algorithms used to find this time point are called timing recovery algorithms and they are described further in Section 4.8.2.

4.4.3 Inter-Symbol Interference

One important property of the pulse shaping filter is that it should avoid interference between symbols. This property for the raised-cosine filter is shown in Fig. 4.8. The dashed curve which is the sum of all five pulses has exactly the same values as each pulse at positions \( kT, k \in \mathbb{Z} \). If the dashed curve is the received signal and it is sampled at positions \( kT \), the sampled values (circles in the figure) will correspond to the correct values for all pulses. If the timing is not correct, the sampled values could be completely wrong. The correct timing in this case means sampling the signal at the exact time points \( t = kT \). To get the right timing, it is important to synchronize the receiver with the rate and the phase of the incoming symbols. This is described further in Section 4.8.2.

Figure 4.8: Demonstration of inter-symbol interference properties of the RC filter. Each of the colored lines represents a single symbol and the dashed line is the sum of the colored lines.
4.5 RF Modulation

The complex FFT of the baseband signal \( x_{bb} = i(t) + jq(t) \) is the solid spectrum in Fig. 4.9, where \( i \) and \( q \) could be, for example, the components of a QAM-signal. If this signal needs to be modulated on to a higher frequency this could be done as

\[
x_{rf}(t) = x_{bb}(t)e^{j2\pi f_0 t}
\]

\[
= (i(t) + jq(t))\left(\cos(2\pi f_0 t) + j\sin(2\pi f_0 t)\right)
\]

\[
= i(t)\cos(2\pi f_0 t) + jq(t)\cos(2\pi f_0 t) + i(t)j\sin(2\pi f_0 t) - q(t)\sin(2\pi f_0 t)
\]

\[
= \left(i(t)\cos(2\pi f_0 t) - q(t)\sin(2\pi f_0 t)\right) + j\left(q(t)\cos(2\pi f_0 t) + i(t)\sin(2\pi f_0 t)\right)
\]

\[
\text{Re}[x_{rf}(t)]
\]

(4.7)

according to [10]. The dashed spectrum shown in Fig. 4.9 is then the complex FFT of \( x_{rf}(t) \). As seen in (4.7) this complex signal contains both \( i \) and \( q \). This signal uses only half the bandwidth compared to the signal

\[
x_{rf}(t) = \text{Re}[x_{rf}(t)],
\]

(4.8)

shown in Fig. 4.10, which according to (4.7) also contains both \( i \) and \( q \). The main difference, besides the different bandwidths, between these two signals is that \( x_{rf} \) is complex valued and is impossible to transmit as one signal. While \( x_{rf} \) is a real valued RF signal and could be transmitted as one signal, in a cable or over the air. If the complex signal \( x_{rf}(t) \) is supposed to be demodulated, this could be done as

\[
x_{rf}(t)e^{-j2\pi f_0 t} = \left(i(t) + jq(t)\right)\left(\cos(2\pi f_0 t) + j\sin(2\pi f_0 t)\right)e^{-j2\pi f_0 t} = x_{bb}(t).
\]

(4.9)

In Section 5.1.6, there is a description of how RF modulation is used in this thesis.

To reach a high frequency, a high sample rate is required. This can be a problem in digital communication systems where the clock frequency and sample rate of DACs and ADCs usually set the limit. One approach is to perform baseband operations such as source/channel coding, mapping, baseband modulation and pulse shaping (described in earlier sections of this chapter) at a low sample rate due to the complexity. The signal is then interpolated to perform the RF modulation at higher sample rates. This can also be done in several steps with different sample rates. The same approach can be used at the receiver by implementing RF demodulation at a high sample rate and then decimate to a low sample rate to perform baseband operations. Another maybe more classic approach is to perform the RF modulation and demodulation with analog components. Sometimes a combination of these two approaches is the best solution. Since the implementation will be done all digital on the SDR14 device, this analog approach is not investigated at all.
4.6 Channel

The channel in this context is where the signal is transferred. This can for example be over the air or in a wire. To be able to determine how to design the transmitter and the receiver, a model for the channel is needed which describe its properties. There exist several different mathematical descriptions of channel effects such as, noise and interference, see more on these in [8]. The most commonly used channel model is, according to [9], the additive white Gaussian noise (AWGN) channel model. This is noise with constant power spectral density (white noise) and the probability density function for the amplitude is a Gaussian distribution. Additive means that it is added to the signal.

4.7 Multiple Access Techniques

It is often the case that a channels capacity is divided among several users. This requires some form of separation among the different users. There exists several ways to do this. Three widely used techniques are time-, frequency- and code division multiple access which are described in the sections below. More about each technique can be found in [9].
4.7.1 TDMA

Time division multiple access (TDMA) is quite easy to understand. This is what we use when, for example, two people speak to each other (in most cases). The first person speaks first, then the other and then the first again, taking turns. It is the same when used in digital communication. Each user of the shared channel gets a time slot for transmission at a certain time interval. This requires good synchronization between senders to not collide, or a good protocol for handling collisions. There exists several protocols for these problems but this is beyond the scope of this thesis.

4.7.2 FDMA

Frequency division multiple access (FDMA) means that each user is assigned a certain frequency range (frequency band or channel) to transmit on. The signal is modulated onto this frequency as described in Section 4.5. This is practical for applications where a continuous data stream is sent, for example radio and TV transmissions where each channel resides at a certain frequency channel. It is important to filter the transmission correct to avoid interference on adjacent frequency bands or channels, for example by using pulse shaping as described in Section 4.4.

4.7.3 CDMA

Code division multiple access (CDMA) is a technique used in broad band modulation to transmit several messages in the same frequency band without the use of frequency- or time multiplexing. Each message that is to be sent is modulated with a so called chip-code. The chip-code has a higher rate than the actual data. This higher rate will spread the signal in the frequency spectrum. Since the spectrum of the signal is wider, the SNR usually becomes higher, especially in cases with distortions of narrow-band type [6]. One way to do this modulation is a simple XOR operation between the code and the data. The new signal will consist of more bits than before because of the higher data rate. These new bits are usually called chips. The same sequence is then used at the receiver to demodulate the signal. This chip-code is unique for all users in a shared channel and it is common that this code is a pseudo random binary sequence (PRBS) [11].

CDMA as a modulation technique should not be mixed up with the mobile phone standards cdmaOne, CDMA2000 and W-CDMA, but the CDMA technique is used in all three of these standards.
4.8 Synchronization

In digital communication, as well as in analog communication, it is important that the transmitter and the receiver are synchronized so that the information sent can be interpreted correctly. In this section two types of synchronization are discussed: carrier and timing synchronization. The synchronization discussed is on waveform level and not on a bit stream level. At the bit stream level, additional synchronization is required to handle bytes, words, frames and so on. Since the synchronization problem is a central part of this thesis, this area is described more thoroughly than the rest of the digital communication system.

4.8.1 Carrier Recovery

When converting a signal down from higher frequencies, it is important that the locally generated sinusoidal signal has the correct phase and frequency compared to the carrier in the signal. Figure 4.11 shows the result of a phase offset at the receiver for a QPSK signal, resulting in a rotated constellation. A frequency offset will result in a constantly rotating constellation as shown in Fig. 4.12. To make things even harder, the phase offset can vary over time depending on, for example, transmission delay. The frequency offset can also vary due to, for example, Doppler effects when the distance between the transmitter and receiver changes [9].

![Figure 4.11: Phase offset at receiver mixer with QPSK signal. Red x marks the four correct points. Blue points are the received values.](image)

There exist many ways to recover the carrier. They can be divided into two main groups: decision-directed (DD) methods which rely on demodulated data and non-decision-directed (NDD) methods which does not. A downside with DD methods is that the signal needs to be good enough to be properly demodulated in order to achieve a correct carrier recovery.

The central part of most methods is the phase-locked loop (PLL) which consists of a phase detector, a loop filter and an oscillator with a variable phase offset. A
Figure 4.12: Frequency offset at receiver mixer with QPSK signal.

simple example of a basic PLL is shown in Fig. 4.13 where the phase detector is the multiplier followed by the low-pass filter. If the input signal is assumed to be \( r(t) = \cos(\omega_i t + \theta_i) \) and the locally generated signal is assumed to be \( s(t) = \sin(\omega_i t + \theta_0) \), then the result of the multiplication is

\[
e(t) = r(t)s(t) \\
= \cos(\omega_i t + \theta_i) \sin(\omega_i t + \theta_0) \\
= \frac{1}{2} \sin(\theta_0 - \theta_i) + \frac{1}{2} \sin(2\omega_i t + \theta_i + \theta_0).
\]

(4.10)

To remove the double frequency component, \( e(t) \) is passed through a low-pass filter and the result is

\[
e_{LP}(t) = \frac{1}{2} \sin(\theta_0 - \theta_i).
\]

(4.11)

If a small phase offset is assumed, the approximation

\[
e_{LP}(t) = \frac{1}{2} \sin(\theta_0 - \theta_i) \approx \frac{1}{2}(\theta_0 - \theta_i)
\]

(4.12)

can be done. This represents the phase error and is passed through a loop filter to generate a correction that is more stable over time. The output from the loop filter is then used as input to the local oscillator which can control the phase and/or frequency of the generated signal \( s(t) \). More on this is described in [10].

A special case of the PLL is the Costas loop. This is used when the carrier is suppressed in the signal and not distinct or at a separate channel. An example of a Costas loop for BPSK from [9] is shown in Fig. 4.14. The input signal is modulated with a message \( m(t) \) in the transmitter.

For QPSK, this has to be extended to the architecture in Fig. 4.15 which is called Spilkers loop [12]. In this case the \( I \) and \( Q \) parts of the signal are modulated by the two messages \( m_1(t) \) and \( m_2(t) \).
4.8 Synchronization

**Figure 4.13:** Basic PLL principle with reference signal $r(t)$ and locally generated signal $s(t)$.

**Figure 4.14:** Costas loop for BPSK carrier tracking.

**Figure 4.15:** Spinkers loop for QPSK carrier tracking.

**Decision-Directed Methods**

The basic functionality of DD methods is to measure the phase error between a received value and the correct value, and correct for this. For example, the error
between the received blue dots and the correct red crosses in Fig. 4.11. This can, for example, be done by using the arctangent operation on $Q$ to get the angle and compare this to the correct angle, $(2n - 1) \cdot \pi / 4$ for QPSK. An example of this can be found in [13].

Even though this sounds like a basic task it could be very hard due to the fact that the correct point of each received symbol needs to be known. In Fig. 4.11 this is trivial in the case of phase errors up to $\pi / 4$, since there are only four available points. However, in a higher-order modulation like 64-QAM this is not that easy. As seen in Fig. 4.16 it could be quite difficult to determine the correct point for a received symbol. One way to solve this could be to only rely on a subset of symbols in the phase error estimation. Using only the four corner-symbols in QAM makes it as easy as in the QPSK case. This affects both the complexity, since a decision has to be made to use a symbol or not, and the speed of phase lock since correction is not made for every symbol.

For QPSK and QAM there is a $\pi / 2$ phase ambiguity which means that phase errors larger than $\pi / 4$ results in an incorrect phase lock. This ambiguity can be solved by using differential modulation as described in [9]. Differential modulation is performed by representing the data with the relative phase compared to the previous symbol instead of the absolute phase of each symbol. This technique can be applied for both M-ary PSK and M-ary QAM and is called differential PSK (DPSK) and differential QAM (DQAM). For DQAM, this can be applied to the amplitude as well and is then called quotient QAM (QQAM) [14]. A downside with differential modulation is that if one symbol is incorrect, this affects both the difference on that and the previous symbol as well as the difference of that and the next symbol. This will result in a twofold increase in bit error rate compared to individual demodulation of symbols.

Figure 4.16: Phase offset at receiver mixer with QAM signal. Red x mark the 64 correct points.
4.8 Synchronization

Non-Decision-Directed Methods

The NDD methods do not rely on a correct demodulation of the signal. This could make these methods better suited for situations where incorrect demodulation of symbols is very probable. An example of an NDD method is the multiply-filter-divide method for PSK modulation, described as squaring loop in [10]. The principle for this method is to introduce a nonlinearity to extract the carrier from the modulated signal. This can be done by raising the signal to the power of \( M \) where \( M \) is the order of PSK (2 for BPSK, 4 for QPSK and so on). This makes the PLL lock onto the frequency \( M\omega_c \). Then this frequency can be divided by \( M \) again to get \( \omega_c \). An example of this kind of demodulator is shown in Fig. 4.17.

![Figure 4.17: Multiply-filter-divide circuitry for M-ary PSK.](image)

4.8.2 Timing Recovery

The timing recovery problem can be described as sampling the received symbols at the correct time offset and with the correct frequency. The frequency is often known and the problem is therefore to sample at the ideal point within each symbol. As described in Section 4.4.3 this is very important to avoid inter-symbol interference, but also as described in Section 4.4.2 to maximize the SNR in the received signal. Timing recovery includes two operations: Estimate sample offset error and correct for this error. The estimation can be performed by various algorithms mentioned below. The correction can, for example, be performed by selecting different samples at one or several earlier decimation operations. If no such operation exists, an interpolation can be performed to generate more samples to select from.

Early-Late Gate Algorithm

The early-late gate algorithm estimates an error by using two extra samples that are early and late compared to the actual sampling point. By comparing these samples, a timing error can be calculated. In Fig. 4.18 the sampling is done at the ideal, an early, and a late time point. As seen in Fig. 4.18, the amplitude of the extra samples is not the same in the non-ideal cases and it is possible to determine if the sample is early or late depending on the amplitude of these samples. In the
early case the late sample has the highest amplitude and in the late case it is the
early sample that has the highest amplitude. Since this algorithm requires at least
three samples per symbol it is impractical for systems with a high symbol rate.
This method is described further in [10].

![Figure 4.18: Early-late gate algorithm for the ideal, the early and the late case.](image)

**Mueller And Muller Algorithm**

The Mueller and Muller algorithm uses only one sample per symbol but needs
samples from two symbols. A timing error is calculated as

\[ e_n = (y_n \cdot \hat{y}_{n-1}) - (\hat{y}_n \cdot y_{n-1}) \]  \hspace{1cm} (4.13)

where \( e_n \) is the error for symbol \( n \), \( y_n \) is sample of symbol \( n \) and \( \hat{y}_n \) is the decoded
symbol \( n \). This algorithm uses only the minimum of 1 sample per symbol which
is good for fast symbol rates, but it is sensitive to carrier offset and therefore
requires a correct carrier synchronization prior to timing synchronization. This can
be a problem if the carrier recovery depends on correctly demodulated symbols.
Examples of the algorithm for the ideal, the slow and the fast case are shown in
Fig. 4.19. The algorithm is further described in [15].

![Figure 4.19: The Mueller and Muller algorithm for the ideal, the slow and the fast case.](image)

(a) Ideal timing, \( e_n = (-1 \cdot 1) - (-1 \cdot -1) = 0 \).
(b) Slow timing, \( e_n = (-0.6 \cdot 1) - (-1 \cdot 0.8) = 0.2 \).
(c) Fast timing, \( e_n = (-0.8 \cdot 1) - (-1 \cdot 0.6) = -0.2 \).
4.8 Synchronization

Gardner Algorithm

The Gardner algorithm uses an extra intermediate sample between two consecutive symbols to generate an error. This means that at least two samples per symbol are required, which could be a problem if the symbol rate is very high. The error is calculated as

\[ e_n = y_{n-1}(y_n - y_{n-2}), \]  

where the distance between two samples is \( T/2 \). One advantage of this algorithm is that it is, as shown in [16], insensitive to carrier synchronization, which means that symbol recovery can be done prior to carrier recovery. This makes carrier recovery that is dependent on correctly demodulated symbols possible. Examples of the algorithm for the ideal, the early and the late case is shown in Fig. 4.20. The algorithm is further described in [16].

Figure 4.20: Gardner algorithm for the ideal, the early and the late case.
Chapter 5

Digital Communication Demo - Implementation

This chapter describes the implementation of a demo for the ADX IP based on digital communication. First the decisions are motivated based on digital communication systems presented in Chapter 4 and the requirements in Section 1.3.2. Then the available hardware is discussed and finally the implementation is described.

5.1 Motivation and Decisions

To demonstrate the advantages of using SP Devices ADX technology, a suitable digital communication system should be chosen and implemented. The idea is to transmit several digitally modulated signals in a cable. One of the signals is at frequency $f_s/4 - f_0$ and in the same cable another signal is transmitted as well, at frequency $f_s/4 + f_0$. The digitally modulated signal will then be received and demodulated. In the analog-to-digital conversion, the signal at the higher frequency will be folded onto the interesting signal due to the problems discussed in Chapter 2. This should be clearly visible in the constellation diagram of the received signal. An illustration of the described frequency spectrum is shown in Fig. 5.1, where the gray part is the received signal. The digitally modulated signal will be folded to higher frequencies as well, but this effect will not be considered in this implementation.

Based on the requirements described in Section 1.3.2, a system that generates a signal with the same frequency characteristics as a W-CDMA signal is chosen to be implemented. W-CDMA is a very commonly used standard in telecommunication. In the end of 2007, 70% of the commercial 3G networks used W-CDMA [17]. To get the frequency properties of this, the symbol rate should be 3.84 MHz and the symbols shall be pulse shaped by a root-raised cosine filter with a roll-off factor of 0.22 [18]. More about the symbol rate is found in Section 5.1.4 and the pulse shaping in Section 5.1.5. An additional factor that makes this implementation a
good choice is the availability at site of a signal generator capable of generating the same kind of signal. This means that the receiver can be tested with that generator to verify the functionality. A decision was made that all signal processing is done on the FPGA, from generation of random data to received complex symbol points.

The design decisions for each of the digital communication system blocks introduced in Chapter 4 are described below. The most important with the implemented digital communication system is that it should clearly illustrate the advantage of the ADX technology. The system should transmit random data on a physical channel and the same channel should be used by some other unit as well. The transmitted data should not be disturbed by anything else than the distortion introduced by the time interleaved ADC. The system will run on the SDR14 from SP Devices because it is the only digitizer that they develop with both DACs and ADCs.

5.1.1 Source and Source Coding

The source chosen is a PRBS. This is quite easy to implement on an FPGA and since the objective is to demonstrate effects on constellation points, the data can be of random type and not known. However, since the sequence is known it is possible to compare the received symbols with a delayed version of this sequence to get a bit error rate (BER). For that reason a reasonable length of 65,535 states is selected for the PRBS since at least one complete sequence is preferred to obtain for comparison.

The rate of the generated bit stream depends on the desired symbol rate and is discussed in Section 5.1.4.

5.1.2 Channel Coding

Since the final step, to demodulate the received symbols to a bit stream will not be done, no channel coding is implemented. But it should still be taken into consideration when evaluating constellation diagrams that this could be implemented. Error correcting codes would allow a certain degree (which is based on the number of extra bits sent) of bit errors to be corrected.
5.1 Motivation and Decisions

5.1.3 Mapping

The original W-CDMA protocol uses QPSK mapping. To achieve a higher data rate in good conditions, this is extended with support of both 16-QAM and 64-QAM as well in the HSPA standards. The available mappings depend on the revision of the standard, both at the base station and in the device used to connect to the base station. For example, Table 5.1a in [19] lists user equipment capabilities for different device classes. Which of the available mappings to use depends on the current link quality. There are also more features to increase the performance but they do not apply to this implementation. To generate a signal similar to W-CDMA and because of the good quality on the channel, all three of these mappings where implemented.

5.1.4 Multiple Access Techniques

W-CDMA uses chip codes as multiple access technique, as the name indicates. This requires coherent demodulation which means full synchronization between modulation at the transmitter and demodulation at the receiver. For this reason and since only random data are to be transmitted, the chip codes are neglected in this implementation. The symbol rate is instead the same as the chip rate for W-CDMA, 3.84 MSPS [18]. The resulting sequence could be interpreted as sending random data to random users if analyzing it as a chip coded signal. The frequency characteristics should still be the same as a signal modulated by correct chip codes. The clock frequency of the FPGA that will be used in this thesis is 200 MHz. Without any additional hardware this limits the ability to achieve a symbol rate of exactly 3.84 MSPS. The closest integer subdivision of the clock frequency is 200/52 which gives a symbol rate of approximately 3.84615 MSPS. When a sample rate of 3.84 MSPS is mentioned in this report it will in practice be 200/52 MSPS. As described in Section 5.1.5 the system is connected to another system on the FPGA that handle the up and down conversion to RF. That system requires a clock frequency of 200 MHz to work as it is supposed to. Therefore, it is not possible to use an external clock source of 52·3.84 MHz.

To transfer four parallel digitally modulated signals as described in Section 5.1, FDMA is utilized. The four W-CDMA signals will be modulated to four different frequencies and filtered so that the adjacent channels do not interfere with each other.

5.1.5 Pulse Shaping

To comply with the W-CDMA standard, the pulse shaping is an RRC filter described in Section 4.4.1, with $\alpha = 0.22$ and $T \approx 0.26042 \mu s \ (\Rightarrow f = 3.84 \text{MSPS})$, according to [20]. To get the advantages with matched filtering, one filter is used in the transmitter and one in the receiver. It is important that the filters comply with the W-CDMA standards so that the filter in the signal generator match the filter in the implemented receiver. With the parameters described above the
The bandwidth of the generated pulse shaped signal is

\[ T(1 + \alpha) = 3.84 \cdot 1.22 \approx 4.68 \text{ MHz}. \]  

(5.1)

According to [18], the spacing between channels is 5 MHz which means that the specifications of the pulse shaping filter is good enough to have several parallel channels with the spacing described in the standard.

### 5.1.6 RF Modulation

During this thesis, a parallel thesis [21] at SP Devices concerns Software Defined Radio (SDR). The objective of that thesis is basically to move a complex baseband signal with a bandwidth of 20 MHz (at 25 MSPS) to an arbitrary RF and also the other way, to down convert an RF-signal to baseband. These two modules are from here on referred to as a digital up converter (DUC) and a digital down converter (DDC). That implementation is also based on the SDR14. The RFs used by the digital demo implementation are from 0 to 400 MHz. The sample rate out of the transmitter is 1600 MSPS (to the DACs) and 800 MSPS in to the receiver (from the ADCs). This makes this unit perfect to include in the implementation of this thesis to handle the RF modulation and demodulation.

As mentioned earlier, one goal is that the generated signal shall have as high bandwidth as possible. Since the RF modulator developed in [21] has a maximum input bandwidth of 20 MHz, it is possible to transmit four channels with the spacings described in Section 5.1.5 and with the techniques described in Section 4.7.2. To be able to generate four channels and place them as shown later in Fig. 5.7(e) on page 45, the signals have to be complex valued. This is because the four channels have to be generated as shown in Figs. 5.7(a)-(d). If the signal is real valued, the frequency spectrum will be mirrored to negative frequencies.

### 5.1.7 Physical Channel

The physical channel is where the actual data will be transmitted in the analog domain. As described in Section 4.6, there exist many types of channels. To have control of the signal on the channel and avoid unknown noise sources, a coaxial cable is used instead of, for example, the air which would be a lot more complex and require extra equipment like antennas. In this implementation, the distortion on the channel is generated either from the DACs or from another signal generator connected to the same cable. The system is modeled with an AWGN channel in Simulink to determine the performance in the ideal case.

### 5.1.8 Synchronization

As described in Section 4.4.3, it is very important to get a correct symbol timing to avoid inter-symbol interference. Therefore, this kind of synchronization was prioritized and implemented prior to carrier recovery. Because of this, a timing recovery method insensitive to carrier synchronization was needed and for that reason, the Gardner algorithm was selected. The timing recovery is supposed to
only correct for a timing phase error. The symbol frequency is assumed to be correct, or at least correct enough so that a continuous phase adjustment can correct for this.

The carrier synchronization is needed to adjust for a carrier phase offset when the SDR14 is used both as transmitter and receiver. When the external generator is used, there is also a small carrier frequency offset as well as a phase offset. The method used for this is decision-directed because of the good quality of the transmission and the lower complexity of these implementations. A decision was taken to prioritize carrier recovery for QPSK. This means that no decision whether to include a received symbol in the algorithm or not is taken, but all received symbols are taken into consideration. The result is that carrier recovery for 16-QAM and 64-QAM can work in some special cases, but it is not expected to. The motivation for this decision is that the only case where a frequency error is assumed is when using the external generator, and this only generates QPSK signals. When using the SDR14 as both transmitter and receiver, the phase offset is assumed to be constant. This means that QPSK can be used to adjust the carrier phase, then the carrier synchronization can be locked before switching to 16-QAM or 64-QAM.

5.2 Hardware

As mentioned in Section 1.3.2, the transmitter and receiver shall run on an SDR14 generator/digitizer. SDR14 offers a Xilinx Virtex-6 FPGA and this is where the transmitter and receiver are implemented using HDL. The structure of the FPGA is presented in Section 5.2.1. The limiting resource on the FPGA for this implementation is the DSP48E1 blocks. How these blocks affect the implementation is discussed in Section 5.2.2.

5.2.1 The FPGA of the SDR14

In the SDR14 there is logic to handle capture, generation, and processing of signals. This logic is implemented on the Xilinx Virtex-6 FPGA and can be divided into blocks as shown in Fig. 5.2.

The data catcher handles the collection of data from the ADCs. This data is time interleaved and corrected with the help of the ADX IP before passed through to the user logic.

The user logic is where customers can implement whatever logic or signal processing they want to perform the desired tasks. This is also where the modules implemented in this thesis are located. The DDC and DUC are also parts of the user logic. There is some additional hardware available and controlled from the user logic. For example, memories to store waveforms from the ADCs or read to the DACs. There are also signals for trigger and GPIO port handling in the user logic. The trigger is the signals that start the capture of samples from the ADC. The GPIO port is a 9-pin connector on the device where 4 pins can be used as general purpose I/O ports. To control the customizable user logic, 16 input and
16 output registers are available. These can be written to or read from by using the programming interface described in Section 5.3.1.

The framework consists of a control unit for the system and for the communication through USB (there are also versions of SDR14 communicating via cPCI/PXI Express or PCI Express).

Figure 5.2: SDR14 block diagram.

5.2.2 The DSP48E1 Block

A DSP48E1 block, from here on referred to as DSP (digital signal processing) block, basically consists of a 25- by 18-bit multiplier followed by a three-input 48-bit accumulator. For more information, see [22]. In this implementation, these blocks are mostly used for the FIR filters at the decimation and interpolation stages. This means that the properties of these blocks must be taken into consideration when deciding upon, for example, word lengths. An extra bit in the word length can significantly increase the required amount of DSP blocks. An example is an FIR filter with 102 16-bit coefficients used at an interpolation stage from 25 to 200 MHz and generated by CORE Generator. If 2-24 bits are used for the input data, it occupies 7 DSP blocks. But if 25 bits are used instead, the same filter occupies 15 DSP blocks. The reason why the limit is 24 blocks is due to the symmetric architecture described in Section 5.3.4. The addition prior to multiplication will add one bit to the input data word length. Another example is when simply multiplying two signals. In this case, two signals with a width of 25 bits will result in a too long critical path if multiplied because several blocks are used. A lower width of 18 bits makes it possible to use a single block and the problem is solved.
5.3 Implementation/Design

This section describes the implementation thoroughly. First with an overview of the system where the transmitter and the receiver are presented and then the different parts are described in several sections.

The transmitter consists of a baseband modulator and a DUC. The DUC is then connected to a DAC for transmission on the channel. As described in Section 5.1.6, it is the baseband modulator that is implemented in this thesis. At the receiver, the DDC is connected to the time-interleaved ADC through the ADX IP and forwards the down converted signal to the baseband modulator. An overview of the transmitter and the receiver is shown in Fig. 5.3 and Fig. 5.4.

![Figure 5.3: Transmitter.](image1)

![Figure 5.4: Receiver.](image2)

The baseband modulator, shown in Fig. 5.5, first generates random symbols and then maps them to constellation points according to QPSK, 16-QAM or 64-QAM. To reach the sample rate corresponding to the system clock frequency of the FPGA (200 MHz), an interpolation by 52 is done. This is done in two steps, by four and then by 13. The RRC filter and the low pass filter LP13 makes sure that no aliasing occur in these steps. Complex modulation is then done at this rate to move the signal to one of the four channels -7.5, -2.5, 2.5, or 7.5 MHz. The sinusoidal signal that is used in the up conversion is generated by a direct digital synthesizer (DDS). The final step is to decimate the signal to 25 MSPS.

The baseband demodulator, shown in Fig. 5.6, first interpolates the signal to the sample rate corresponding to the system clock frequency of the FPGA. Complex mixing is then done to move the signal from one of the channels back to 0 Hz. The phase of the DDS is controlled by the carrier recovery algorithm. The signal is then decimated by 52 in two steps, by 13 and then by four, to generate a single symbol from 52 samples. The selection of this sample is assisted by the Gardner algorithm for timing recovery. Before the decimation by 13, LP13 is applied and between the decimation by 13 and by four, the RRC filter is applied.

The baseband modulator and demodulator only generates and receives one of the four channels. Thus, to generate four parallel channels, the baseband modulator has to be duplicated four times. The baseband modulators are then set to
generate one of the available channels (-7.5, -2.5, 2.5 or 7.5 MHz) each, and the signals from the four modulators are added together. The result is illustrated in Fig. 5.7(e). The output signal from the baseband modulator, composed of four parallel channels, is then passed on to the DUC, where it is converted to a higher frequency as illustrated in Fig. 5.7(f). There is no use in demodulating more than one channel in the receiver since only one signal can be displayed at once.

**Figure 5.5: Baseband modulator.**

**Figure 5.6: Baseband demodulator.**
5.3 Implementation/Design

Figure 5.7: Frequency spectrum of four channels from four baseband modulator blocks, added together and then modulated by the DUC to 130 MHz.

5.3.1 PC Interface

To be able to interact with SDR14, SP Devices offers the software ADCaptureLab which is described in Section 1.4.1. This software is good for plotting the captured signal and to adjust parameters in the device. But when capturing and displaying the signal is not enough, a more flexible approach is required. Therefore, SP Devices also offers a simple and powerful programming interface, ADQ-API for C/C++. This application programming interface (API) makes it possible to develop customized software interacting with the device. There is a third option as well, which is to use a Matlab script to communicate with the device. Since Matlab is a powerful tool for signal processing this is ideal for that type of tasks. A customized script can then be written for controlling and interact with the device.

In the digital demo implementation, a Matlab script is used on a PC for control of the implemented modules and for displaying the acquired signals. The available control signals are listed in Table 5.2 and correspond to register values set in the user registers. The available signals to acquire are listed in Table 5.1. It is only possible to send two signals, with a sample rate of maximum 800 MSPS, to the PC at a time. Which two signals to send depends on the display control signal. In most cases the two signals are the I part and the Q part of a complex signal.
Digital Communication Demo - Implementation

### Table 5.1: Available signals from SDR14 to PC.

<table>
<thead>
<tr>
<th>Display Value</th>
<th>Signal Sent to PC</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>From ADC after ADX correction.</td>
</tr>
<tr>
<td>1</td>
<td>Demodulated symbols. Preferably displayed in a constellation diagram.</td>
</tr>
<tr>
<td>2</td>
<td>Generated signal from baseband modulator.</td>
</tr>
<tr>
<td>3</td>
<td>Received signal from DDC.</td>
</tr>
<tr>
<td>4</td>
<td>Generated signal from DUC.</td>
</tr>
<tr>
<td>5</td>
<td>Timing and carrier phase offset values.</td>
</tr>
</tbody>
</table>

### Table 5.2: Control signals from PC.

<table>
<thead>
<tr>
<th>Signal</th>
<th>Register[Bits]</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>display</td>
<td>0[2:0]</td>
<td>Selects which signal to output to PC.</td>
</tr>
<tr>
<td>channel_mode</td>
<td>0[10:3]</td>
<td>Selects what modulation to use on each channel, two bits per channel where ’00’=&gt;only zeros, ’01’=&gt;QPSK, ’10’=&gt;16-QAM and ’11’=&gt;64-QAM.</td>
</tr>
<tr>
<td>receive_channel</td>
<td>0[12:11]</td>
<td>Selects which of the four channels to receive.</td>
</tr>
<tr>
<td>receive_scale</td>
<td>0[15:13]</td>
<td>Scale up the input to the baseband demodulator by $2^{\text{receive_scale}}$.</td>
</tr>
<tr>
<td>carrier_recovery</td>
<td>0[16]</td>
<td>Carrier recovery active or not.</td>
</tr>
<tr>
<td>bypass_analog</td>
<td>0[17]</td>
<td>Bypass ADC and DAC or not. This means that the DUC connects directly to the DDC.</td>
</tr>
<tr>
<td>remove_q</td>
<td>0[18]</td>
<td>Q input to DDC = ADC or zero.</td>
</tr>
<tr>
<td>mixer1_freq</td>
<td>1[31:0]</td>
<td>Fine grained frequency to DUC/DDC. Equal steps from -50 to +50 MHz.</td>
</tr>
<tr>
<td>mixer2_freq</td>
<td>2[3:0]</td>
<td>Coarse grained frequency to DUC/DDC. Signed value in 100s of MHz, e.g. ’0010’=&gt;200 MHz.</td>
</tr>
<tr>
<td>gardner_limit</td>
<td>3[23:0]</td>
<td>Limit of integrated error in Gardner algorithm for timing synchronization before changing timing offset.</td>
</tr>
<tr>
<td>cr_limit</td>
<td>4[25:0]</td>
<td>Limit of integrated error in the carrier synchronization algorithm before changing local carrier phase offset.</td>
</tr>
<tr>
<td>cr_step_size</td>
<td>5[31:0]</td>
<td>Step size of changes to local carrier phase offset.</td>
</tr>
</tbody>
</table>

Table 5.1: Available signals from SDR14 to PC.

Table 5.2: Control signals from PC.
5.3 Implementation/Design

5.3.2 Symbol Generation

To generate the symbols, a PRBS of length 16 is used, as explained in Section 5.1.1. This sequence is generated by an linear feedback shift register (LFSR). The feedback polynomial describes which bits to XOR for feedback and is selected so that a maximum-length sequence is obtained. One example of a 16-bit feedback polynomial which generates such a sequence is $x^{16} + x^{14} + x^{13} + x^{11} + 1$ [23]. This is also the polynomial used in this implementation. From this sequence, $n$ bits are used for each symbol ($n=2$ for QPSK, 4 for 16-QAM and 6 for 64-QAM). Lookup tables are then used to translate these bit patterns to complex I and Q values. Before the next symbol is generated the LFSR is shifted $n$ steps. The resulting LFSR with 6 output bits is illustrated in Fig. 5.8.

![Figure 5.8: The LFSR used for symbol generation.](image)

5.3.3 Sample Rate Conversion and Filtering

Since the original sample rate (as well as symbol rate since each symbol starts out as one sample) is 3.84 MSPS and the sample rate out of the baseband modulator is 25 MSPS, there is a need for both interpolation and decimation. The reason why both are needed is since 3.84 is not a factor of 25. At least an interpolation by 13 and a decimation by 2 is required. Interpolation by four and 13, followed by decimation by 8 is chosen. This is to be able to modulate with a carrier wave to the different channels at the system clock rate of 200 MHz. And also to get more samples per symbol for better resolution in timing offset corrections at the receiver since the same steps are taken at the receiver in reversed order. To prevent aliasing and perform pulse shaping, filters with different specifications are used. This section describes the different filters in the context of purpose, requirement, and specifications. All filters are FIR filters. The length of the filters is not optimized but selected as good enough to not introduce clearly visible distortions in the Simulink Model.
RRC Filter

The RRC filter is the first filter in the transmitter chain and the last in the receiver chain. It is applied directly after mapping and before demapping. As described in Section 5.1.5, it should have a roll-off factor of 0.22. Since there is only one sample per symbol this filter also includes an interpolation by four (from 3.84 MSPS to 15.38 MSPS) at the transmitter to be able to get a smooth pulse shaped signal and increase the sample rate. The coefficients for this filter are generated and exported from the corresponding block in the Simulink Model. The length of the filter is 65 coefficients. The magnitude response of the filter is shown in Fig. 5.9. The filters at the transmitter and the receiver have the same specifications to perform matched filtering, as described in Section 4.4.2.

Interpolation/Decimation by 13 - The LP_{13} Filter

When upsampling by 13, from 15.38 MSPS to 200 MSPS at the transmitter, spectral replications appear at $k15.38$ MHz where $k = \pm 1, \ldots, \pm 12$. Since the signal from the RRC filter has a bandwidth of 5 MHz centered at 0 Hz, there is
a need for a lowpass filter with a pass band of 0-2.5 MHz and a stop band from about 12.88 MHz (=15.38 - 2.5). 12.88 MHz is rounded down to 12.5 MHz.

At the receiver, when downsampling by 13 from 200 MSPS to 15.38 MSPS, spectral components outside ±7.69 MHz will introduce distortion due to aliasing. Four channels exist in parallel at this point with a total bandwidth of 20 MHz centered around 0 Hz, as illustrated in Fig. 5.10(a). Before downsampling, mixing is done to center the 5 MHz channel that should be demodulated at 0 Hz as described in Section 5.3.5 below, for example as illustrated in Fig. 5.10(b). This means that after mixing, spectral components within 2.5 MHz from ±15.38 MHz will introduce aliasing to the channel at 0 Hz. In Fig. 5.10, the case of mixing by ±7.5 MHz is illustrated. Channel D will end up at 15 MHz, introduce aliasing at -2.5 - 2.5 MHz, and thus filtering is required. So, just like in the case of the interpolation filter, a pass band of 0-2.5 MHz and a stop band from 12.5 MHz is required.

Because of equal specifications, the same filter is used for both interpolation and decimation by 13. This filter is designed in Matlabs Filter Design & Analysis Tool with the equiripple method. The length of the filter is 45 coefficients. The magnitude response of the filter is shown in Fig. 5.11. To achieve the same signal levels after the filter as before the filter at interpolation, the coefficients are multiplied by 13 for the filter in the transmitter.

Figure 5.10: Frequency spectrum of four channels (A, B, C, and D) demonstrating aliasing when decimating by 13 after mixing with 7.5 MHz.
Interpolation/Decimation by 8 - The LP₈ Filter

When downsampling by 8 from 200 MSPS to 25 MSPS at the end of the transmitter, spectral components outside ±12.5 MHz will introduce aliasing. At this point four channels exist that are centered around 0 Hz. This means that no spectral components exist outside ±10 MHz and therefore filtering is not required before this downsampling.

When upsampling by 8 from 25 MSPS to 200 MSPS at the receiver, spectral replications appear at k·25 MSPS where k = ±1, . . . , ±7. The only operation between this and the LP₁₃ filter is a mixer. Because of that, and the fact that the LP₁₃ filter removes spectral components which would introduce aliasing in the downsampling by 13 later on, no filtering is required after the upsampling. However, the LP₈ filter can relax the requirements on the LP₁₃ filter. This is a good thing since the LP₈ filter at the receiver also interpolates by 8 and therefore is cheaper than the LP₁₃ filter which operates at full speed, see Section 5.3.4 on polyphase decomposition. The LP₈ has 101 coefficients but still only requires 8 DSP blocks compared to the LP₁₃ which requires 23 DSP blocks and only has 45 coefficients. Another advantage is if more than one channel should be received,
the LP₈ filter can be applied once at the receiver while the LP₁₃ filter has to be applied once for each parallel received channel. These two reasons are why this filter is included even though it might not be needed.

Since four channels exist from -10 MHz to +10 MHz, and the spectral replications appear at \( k \times 25 \text{ MSPS} \) where \( k = \pm 1, \ldots, \pm 7 \), a pass band of 0-10 MHz and a stop band from 15 MHz is required. The magnitude response of the filter is shown in Fig. 5.12. To achieve the same signal levels after the filter as before the filter, the coefficients are multiplied by 8.

![Magnitude Response (dB)](image)

Figure 5.12: Magnitude response of the LP₈ filter implemented in hardware.

### 5.3.4 Mapping Filters to Hardware

To map the filters to hardware, Xilinx CORE Generator is used. The filter coefficients are exported from Matlab and imported to *Xilinx LogiCORE IP FIR Compiler v6.1*. The datasheet for this tool can be found in [24]. The width of the input signal and the coefficients are described in Section 5.3.8. To decrease the resource utilization, especially in terms of DSP blocks, the FIR blocks make use of the fact that the filters are symmetric and in some cases operate at different sample rates.
Symmetric filters means that for a filter of length $L$, coefficient 1 and $L$, 2 and $L-1$, and so on are identical. An example of a symmetric filter is the LP$_8$ filter shown in Fig. 5.13. This means that every such pair of coefficients can share a multiplier and these samples can be added together prior to multiplication. This symmetric architecture will only require half the number of multipliers compared to the case of unique coefficients and direct-form implementation. This is described in [25]. The direct-form implementation is shown in Fig. 5.14 and the symmetric form with reduced number of multipliers is shown in Fig. 5.15.

When interpolating or decimating by $L$, it is unnecessary to perform multiplication on all samples. In the case of decimation, only every $L$th value will be used, and in case of interpolation, only every $L$th value is non-zero after the upsampling. This is utilized in filters called polyphase filters and makes it possible for such an interpolation or decimation filter with a factor of $R$ to run $R$ times faster. If resource usage is the goal instead of speed, multipliers can be shared and only $L/R$ multipliers will be used. The techniques are further described in [26].

According to [24], both polyphase decomposition and symmetric properties are used in FIR Compiler v6.1 to be able to minimize the area. The technique used to combine this is described in [27].

![Figure 5.13: Impulse response of the LP$_8$ filter.](image-url)
5.3 Implementation/Design

Figure 5.14: Direct-form FIR filter.

Figure 5.15: Symmetric FIR filter with reduced number of multipliers.

5.3.5 Baseband Mixer and DDS

As described earlier, this system has an output bandwidth of 20 MHz at 25 MSPS consisting of four channels at -7.5/-2.5/2.5/7.5 MHz. To convert a channel from 0 Hz to any of these frequencies (or back to 0 MHz), a complex multiplication by a complex exponential, as in (4.7), has to be performed. Since

\[ e^{j2\pi f_0 t} = \cos(2\pi f_0 t) + j\sin(2\pi f_0 t), \]

the complex exponential could be replaced by two direct digital synthesizers (DDS), one generating the cosine wave and one the sine wave. The DDS is a custom IP block generated by CORE Generator and it is simply a look-up table with values for the sinusoidal signal. A counter is used to calculate the address for the lookup table, and the increment-value for the counter is the value that controls the frequency of the generated waveform. A high value will generate a higher output frequency compared to a lower value. The actual baseband mixer is a complex multiplier generated by CORE Generator and is used as shown in Fig. 5.16.
At the transmitter, the phase and the frequency of the DDS is constant but at the receiver both of these are variable. The reason why the frequency is variable is to be able to down convert different channels with the use of the same DDS. It is set to either 2.5 or 7.5 MHz. The negative frequencies are generated by simply negating the value from the DDS that generates the sine wave. The reason for a variable phase offset at the receiver is to be able to compensate for the delay between the transmitter and the receiver, see Section 5.3.7 about carrier recovery.

![Baseband mixer implementation](image)

Figure 5.16: Baseband mixer implementation.

### 5.3.6 Timing Recovery

The Gardner algorithm for timing recovery is implemented according to (4.14). The error is generated from I and Q individually, and the sum is used as the error. The challenge is to generate a stable and correct adjustment based on these error calculations. This is the task of the loop filter.

The timing adjustment is made at the two decimation operations in the baseband receiver as seen in Fig. 5.6. Since there are two decimation steps, one by 13 and then one by four, there is a total of 52 steps available for the timing adjustment. To select the correct sample at these two points, two control signals are used. They are described in Table 5.3. Because of the few number of steps

<table>
<thead>
<tr>
<th>Timing Offset Value</th>
<th>select_4</th>
<th>select_13</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>:</td>
<td>:</td>
<td>:</td>
</tr>
<tr>
<td>12</td>
<td>0</td>
<td>12</td>
</tr>
<tr>
<td>:</td>
<td>:</td>
<td>:</td>
</tr>
<tr>
<td>13</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>:</td>
<td>:</td>
<td>:</td>
</tr>
<tr>
<td>25</td>
<td>1</td>
<td>12</td>
</tr>
<tr>
<td>:</td>
<td>:</td>
<td>:</td>
</tr>
<tr>
<td>26</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>:</td>
<td>:</td>
<td>:</td>
</tr>
</tbody>
</table>

Table 5.3: Timing adjustment.
available, the loop filter needs to filter the error. There is also a problem that the error is proportional to the amplitude of the signal. Then, if the amplitude of the signal is not known, it is hard to make use of a proportional regulation on the timing phase.

The designed loop filter was implemented as an integration of the error. When it reaches a value that is high or low enough, \texttt{gardner\_limit}, the output of the filter increases or decreases one step and the accumulated error is reset. This can be implemented with just addition and logic. These values are chosen quite high which results in a slower regulator but this is not a problem because the timing offset is expected to be constant. It is a bigger problem if the timing is adjusted to fast which result in a very unstable regulation. There is also a delay of eight symbols because the adjustment is partially made before the RRC filter which limits the speed of the regulator. The tunable parameter in this filter is the value of \texttt{gardner\_limit}. Which value for this parameter that is the best is discussed in Section 6.4.

5.3.7 Carrier Recovery

The decision-based carrier recovery means that the symbol is first decoded and then a phase error is generated. The phase error is estimated according to

\[ e_n = \text{sgn}(Q_n)I_n - \text{sgn}(I_n)Q_n \]  

where \texttt{sgn()} is the signum function. This correspond to the phase error generation in Fig. 4.15. The resulting error is the same as the arctangent operation, in cases when the amplitude of \( Q_n \) and \( I_n \) is one. The downside with this solution is that the size of the error will be proportional to the amplitude of the signal. But the upside is that the arctangent operation is not needed.

This error is filtered by the same loop filter as the timing recovery, described in Section 5.3.6. Since only a phase error is present when the SDR14 is used both as transmitter and receiver, the output of the loop filter is adjusting the phase of the DDS generating the signal to the final mixer. But a constant adjustment of the phase gives the same result as adjusting the frequency. A difference in the loop filter compared to the one used in timing recovery is that the resolution of the carrier phase offset is significantly larger than the 52 steps available in the timing phase offset. This gives a second tunable parameter: the step size of the adjustments. How to set these parameters to achieve the best result is discussed in Section 6.3.
5.3.8 Word Lengths

In the system, fixed-point arithmetic is used. The word lengths chosen could affect the performance in a bad way if chosen poorly, i.e., too short. The word lengths chosen are based on the word lengths to the DUC and from the DDC which is 16 bits, and thus sets the lower limit. Due to the DSP blocks response on different word lengths discussed in Section 5.2.2, the word length is chosen to 24 bits throughout the system. This will clearly pass the lower limit of 16 bits, and 24 bits should still only introduce a small amount of extra hardware compared to 16 bits.
Chapter 6

Digital Communication
Demo - Tests and Results

Even though the transmitter and the receiver will not be used to generate and receive meaningful data, the performance is important. A measure of the bit error rate (BER) might not be interesting, but a rotating constellation diagram does not appear very convincing when you try to illustrate the performance of an algorithm. Therefore, algorithms for timing and carrier recovery had to be implemented and tested. This chapter will describe and show the results from testing the digital communication demo.

6.1 ADX IP Demo

This section will describe tests and evaluation of how well the digital communication demo implementation performs the task of demonstrating the performance of the ADX IP. To test if the distortion from the interleaving of ADCs will affect the constellation diagram, this was first tested in the Simulink model. One way to test this is to generate a sinusoid at the frequency where the distortion spur will appear and add this to the signal between the baseband modulator and demodulator. This test showed that the effects where clearly visible, as shown in Fig. 6.1.

This test was performed in hardware as well. To do this, a real interleaving distortion was introduced with the test setup shown in Fig. 6.2. To illustrate that the blocker sinusoid signal only affect the frequencies described in (2.2) and (2.1), the transmitter is configured to send on two channels with QPSK modulated data. The signal generator is configured to send a sinusoid at 292.5 MHz. Figure 6.3 shows the signal from the ADCs, with the spur from the interleaving distortion clearly visible in the middle of the channel at 107.5 MHz. The constellation diagrams for these two channels are shown in Fig. 6.4. It is very easy to see the distortion introduced by the blocker signal. In this case, the ADX IP was deactivated and Fig. 6.5 shows the same plots but with the ADX IP activated. It is
clearly visible in Fig. 6.5(b) that the spur is not affecting the signal any more. The special formation of circles at each of the symbols is the same for both 16-QAM and 64-QAM as well. The size of this circle depends on the amplitude of the blocker sinusoid.

Figure 6.1: Simulated interleaving distortion in Simulink.

Figure 6.2: Test setup.
Figure 6.3: Signal from the TI-ADC, ADX is deactivated. The interleaving tone generated from the sinusoidal is clearly visible at 107.5 MHz. Two channels are used on 92.5 and 107.5 MHz.

Figure 6.4: Constellation diagrams for the two channels in Fig. 6.3 with ADX deactivated.
(a) 92.5 MHz.

(b) 107.5 MHz.

Figure 6.5: Constellation diagrams for the two channels in Fig. 6.3 with ADX activated.
To evaluate how suitable the implemented system is as a transceiver, three factors were evaluated. The first factor is how robust the system is for noise. This was evaluated by applying AWGN in the Simulink model. The second factor is how well two adjacent channels interfere with each other and the third is how well the system complies with the W-CDMA standard. But first, a test to show that the three implemented mapping techniques really work is done. Three channels are used: One with QPSK, one with 16-QAM and one with 64-QAM. The result is shown in Fig. 6.6.

Figure 6.6: Test of the available mappings.
6.2.1 AWGN in Simulink

In this test, AWGN was added on the signal between the baseband modulator and the baseband demodulator. The power of the AWGN was increased until it was clear that symbols would be demodulated as adjacent symbols. The average power of the signal and the AWGN is compared to calculate the SNR. According to [26], this is calculated as

\[
\text{SNR} = \frac{\text{Signal power}}{\text{Noise power}} = \frac{\sum_{n=0}^{N-1} |x_s(n)|^2}{\sum_{n=0}^{N-1} |x_r(n)|^2}.
\]

(6.1)

The power of the signal was measured when only transmitting on one of the four channels. The result was that the system could handle an SNR of about 3 dB. In Fig. 6.7, two tests with QPSK modulated data are shown, one with an SNR of 3 dB that passes the test and one with SNR of 0 dB that fails the test. These two limits are only approximated, in order to get an idea of the performance. Unfortunately, this is not tested in hardware but it would be really interesting to be able to compare these results to real-world results. This test could be performed to determine the requirement on the SNR for both 16-QAM and 64-QAM as well, but was not since it would have been harder to evaluate the results visually.

![Figure 6.7: AWGN test in Simulink with SNR of 0 dB and 3 dB. Fails for 0 dB since some symbols are too close to the other symbols.](image)

6.2.2 Inter-Channel Interference

One requirement for a transmitter with several channels in the frequency domain is that they are filtered and separated enough to not interfere with each other. According to the calculations in Section 5.1.5, 5 MHz should be enough to not...
interfere with each other. This is tested by visually inspecting the FFT of the four channels sent with one of the center channels deactivated. The result is shown in Fig. 6.8 and one can see that the power in the empty channel is about 40 dB lower than the power of the signals sent. A test was also made to receive a channel both when adjacent channels are silent and when they are used. The result is shown in Fig. 6.9, and it can be seen that adjacent channel does not really affect the received signal more than very little.

![Figure 6.8: FFT plot showing inter-channel interference.](image)

![Figure 6.9: Test of inter-channel interference, constellation diagrams.](image)
6.2.3 W-CDMA Compliance

Since the digital demo application design is based on parts of the W-CDMA standard, it should also comply with this standard in the parts that are used, which is the symbol rate, pulse shaping, and mapping method. To test this, a signal generator capable of generating a signal with these specifications is used to generate a signal that is received by the implemented receiver and demodulated. The signal generator is a Rhode & Schwarz SMIQ03B. To compensate for the small error of symbol rate in the receiver, described in Section 5.1.4, the signal generator was set to a symbol rate of approximately 3.8461538 instead of 3.84. More on how well the receiver performs when this is not compensated for is discussed in Section 6.3. Figure 6.10(b) shows a constellation diagram of the demodulated symbols and it can be seen that it is just as good results as in the case when the transmitter of the demo implementation is used.

6.3 Carrier Recovery

The Rhode & Schwarz signal generator can, as discussed in Section 6.2.3, generate a signal almost identical to the one of the QPSK channels generated from the transmitter developed during this thesis. When trying to receive this signal, the performance of the carrier recovery algorithm is important. This is because of the different clock sources in the receiver and the signal generator. These clocks will differ somewhat from each other and the difference may even vary over time which will result in both a carrier frequency error and a phase offset of that frequency.

A difference in frequency will result in a rotating constellation diagram. By adapting to the frequency of the received signal this could be avoided. A frequency offset is the same as a constantly increasing or decreasing phase offset. Due to the implementation of the carrier recovery algorithm, described in Section 5.3.7, the ability to adapt to frequency offsets is limited by the speed of the phase offset correction.

It turned out to be quite hard to trim the frequency of the transmitted signal to match the frequency in the receiver, since an adjustment of the frequency also affected the symbol rate. Tests show that a too large frequency offset is not possible to compensate for, but the algorithm has no problems handling smaller frequency offsets. During testing, the results show that frequency offsets up to approximately 2 kHz is possible to handle which is about 0.002% at 100 MHz. Figure 6.10 shows a frequency offset in the case of carrier recovery activated or deactivated.

As described in Section 5.3.7, both the limit of the accumulated carrier phase error before adjustment and the size of these adjustments can be tuned. This is experimented with, and the conclusion is that it is very dependent on the input signal. A signal with larger amplitude requires a larger limit on the accumulated error to keep the phase stable, and a larger frequency error requires larger adjustments to keep up. Thus, the conclusion for these parameters is that a larger limit and a small step size generate a more stable adjustment, but is dependent on a frequency in the transmitter that is close to the one in the receiver.
6.4 Timing Recovery

As described in Section 4.4.3, it is very important to achieve correct timing. This can clearly be seen in Fig. 6.11, where the timing offset value, described in Section 5.3.6, is set to three different values.

As described in Section 5.3.6, there are 52 offset values available. As seen in Fig. 6.11, only a few steps wrong in the offset is clearly visible. This is observed in the tests and could be a limiting factor on the performance of the implementation. The risk that the optimal offset value is between two of the available values is very probable. This can be observed as the Gardner algorithm often does not stay at the same offset value, but toggles between two of them. The case is also sometimes that one of them is much better. But since it still generates a small error, it will eventually change to the other one and then back again.

The parameter of the Gardner algorithm that sets the limit of the accumulated timing error before making an adjustment has the same property as the limit in the carrier recovery case. A too low value will make the timing offset vary considerably and thus result in a noisy demodulation. Therefore, the conclusion is to select a large value since the timing offset is assumed to be constant.
Figure 6.11: Simulink simulation of different timing offset values.

6.5 Resource Utilization

Since the critical resource of the FPGA has been the DSP block, only this has been included in the results. The utilization of DSP blocks is presented in Table 6.1. As seen in this table, the utilization is less than 43 %.

<table>
<thead>
<tr>
<th>Unit</th>
<th>DSP blocks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Framework</td>
<td>74</td>
</tr>
<tr>
<td>DUC</td>
<td>74</td>
</tr>
<tr>
<td>DDC</td>
<td>50</td>
</tr>
<tr>
<td>BB Modulator</td>
<td>4 · 13</td>
</tr>
<tr>
<td>LP₁₃</td>
<td>3 + 3</td>
</tr>
<tr>
<td>Mixer</td>
<td>3</td>
</tr>
<tr>
<td>RRC</td>
<td>2 + 2</td>
</tr>
<tr>
<td>BB Demodulator</td>
<td>75</td>
</tr>
<tr>
<td>LP₁₃</td>
<td>23 + 23</td>
</tr>
<tr>
<td>LP₈</td>
<td>8 + 8</td>
</tr>
<tr>
<td>Mixer</td>
<td>3</td>
</tr>
<tr>
<td>RRC</td>
<td>4 + 4</td>
</tr>
<tr>
<td>Gardner</td>
<td>1 + 1</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td>325 of 768</td>
</tr>
</tbody>
</table>

Table 6.1: Utilization of DSP blocks on the FPGA.
Chapter 7

Conclusions

This chapter concludes the thesis by presenting the final implementations with a discussion of the results compared to the requirements. Finally, some ideas for future work that did not fit into the time frame of this thesis are proposed.

7.1 Analog Video Demo

The analog video demo was implemented by combining commercially available products and the SDR14 device. The video signal used was a composite video signal, modulated to a frequency of 55.25 MHz (analog TV channel 3 in western Europe). This was combined with a sinusoidal signal at a much higher frequency and sampled with the TI-ADC in the SDR14 device to digital values. When sampling, due to mismatch errors in the TI-ADC and the selection of the higher frequency according to (3.1) on page 15, the higher frequency component was folded onto the video signal. The signal was converted back to an analog signal by the DAC in the SDR14 and displayed with the help of a TV tuner USB device on a PC. By activating SP Devices’ ADX IP, the mismatch effects in the TI-ADC are compensated for and the higher frequency component is not folded onto the video signal. The effects of the introduced distortion is shown in Fig. 3.5 on page 18, and the image of the same video signal when using SP Devices’ ADX IP is shown in Fig. 3.6. By looking at these images, it should be clear for any person which one is the best. This fulfills the requirement of a clear demonstration of how the algorithms improves the operation in the analog video case. The use of composite video and a real-world analog TV channel should make it easy to relate to real applications and thus fulfills that requirement as well. It should also be clear that the case of several signals transmitted at different frequencies in the same medium is very common. The simplest example is probably the air where TV, radio, mobile phones etc. utilize different frequencies. Therefore, it should be possible, for a person without any particular knowledge in signal processing, to tell that the higher frequency signal should not interfere with the video signal.
7.2 Digital Communication Demo

The digital communication demo was implemented on the FPGA of the SDR14 device. The device was used by implementing the transmitter and the receiver on the same unit and the signal is transmitted through an analog cable. The signal consists of parallel W-CDMA channels, further described in Section 7.2.1. The signals are generated and received in the baseband between -10 MHz and 10 MHz, at a sampling rate of 25 MSPS. To make use of the negative frequencies, the signal is complex with an I part and a Q part. To be able to transmit the signal through the cable, the signal is modulated to a radio frequency before digital-to-analog conversion, and back again after analog-to-digital conversion. This is done by a digital up converter (DUC) module and a digital down converter (DDC) module, both developed in a parallel thesis at SP Devices [21] and instantiated on the same FPGA as the transmitter and receiver. The effects of the introduced distortion on the transmitted signal could easily be seen in the constellation diagrams. This is shown in Fig. 6.4 on page 59, and with compensations for mismatch effects shown in Fig. 6.5. Thus, the requirement on demonstration of constellation diagram effects is fulfilled.

7.2.1 W-CDMA

The communication system selected is based on the W-CDMA standard. The specifications taken into consideration was a symbol rate of 3.84 MHz, mapping according to QPSK, 16-QAM and 64-QAM, and pulse shaping with a root-raised cosine filter with a roll-off factor of 0.22. A symbol rate of exactly 3.84 MHz was not achieved due to the FPGA clock frequency of 200 MHz required by the DUC and DDC modules. Instead, a symbol rate of $\frac{200}{52} \approx 3.8465$ was used. To test the module, a signal generator capable of generating an emulated W-CDMA signal with QPSK modulation, the specified pulse shaping and the implemented symbol rate, was used. The implemented receiver could correctly demodulate this signal which fulfills the requirement on the frequency properties of a known and common communication standard.

7.2.2 Bandwidth

With a symbol rate of about 3.84 MHz and pulse shaping with a roll-off factor of 0.22, the bandwidth of the W-CDMA signal is about 4.7 MHz. The W-CDMA standard suggests a channel separation of 5 MHz and the bandwidth available in the baseband is 20 MHz. This makes it possible to transmit four parallel channels. This should fulfill the requirement on the high bandwidth. However, even larger bandwidth could be possible as described in Section 7.3.

7.2.3 Carrier Recovery

Carrier recovery is required to be able to achieve the correct frequency and phase offset of the local carrier at the receiver, used for demodulating one of the four received channels. An incorrect frequency and phase will affect the rotation of the
constellation diagram. To perform carrier recovery, a decision-directed algorithm was implemented to calculate the phase error and adjustments were made on the phase of the carrier used for demodulation. The carrier recovery was successful in obtaining an upright constellation diagram for a constant phase error. However, there is a $\pi/4$ phase ambiguity. This could be solved with the use of DPSK as described in Section 7.3. Since a constant phase adjustment can be seen as a frequency adjustment, the carrier recovery could also handle a frequency error of up to about 2 kHz.

### 7.2.4 Timing Recovery

To correctly demodulate the received signal, it has to be sampled at the correct time point within each symbol period. This is controlled by the timing recovery algorithm. In this implementation, the Gardner algorithm was used for calculating the timing offset error. The corrections were done in the two decimation stages of 13 and four in the receiver, resulting in a total of 52 sampling positions to choose from. The timing recovery was successful, but the 52 samples is probably limiting the performance of the system, i.e. the scattering of the constellation points.

### 7.3 Future Work

To improve the digital demonstration, it could be extended with the ability to send real data instead of just random data. In that way, it could be easier to grasp the interference when seeing something like an image be distorted instead of just looking at a constellation diagram. However, as mentioned in Section 7.2.3, the phase ambiguity has to be solved by, for example, implementing differential encoding. Some kind of protocol for the transmission needs to be implemented as well, to be able to detect the start and the end of a transmission.

To be able to achieve a better timing synchronization, the timing adjustments could be integrated into the DDC. In the DDC there are initially, due to the sample rate of 800 MSPS of the TI-ADC, 208 samples to select from for each of the received symbols. This would probably improve the performance of the receiver.

To achieve a higher bandwidth, several DUCs could be run in parallel at different frequencies. If $N$ DUCs are combined with one baseband modulator, one DDC, and one baseband demodulator, this would result in a bandwidth of $20N$ MHz. The $N$ DUCs would be combined by addition, just as the four channels are combined in the current solution. Since it would probably be the number of DSP blocks available on the FPGA that would set the limit, a maximum of 120 MHz should be possible, given the resource usage in Table 6.1. The number of DSP blocks could probably be decreased, enabling an even higher bandwidth, by optimizing the filtering operations.
Bibliography


Upphovsrätt

Detta dokument hålls tillgängligt på Internet — eller dess framtida ersättare — under 25 år från publiceringsdatum under förutsättning att inga extraordinära omständigheter uppstår.

Tillgång till dokumentet innebär tillstånd för var och en att läsa, ladda ner, skriva ut enstaka kopior för enskilt bruk och att använda det oförändrat för icke-kommersiell forskning och för undervisning. Överföring av upphovsrätten vid en senare tidpunkt kan inte upphäva detta tillstånd. All annan användning av dokumentet kräver upphovsmannens medgivande. För att garantera äktheten, säkerheten och tillgängligheten finns det lösningar av teknisk och administrativ art.

Upphovsmannens ideella rätt innefattar rätt att bli nämnd som upphovsman i den omfattning som god sed kräver vid användning av dokumentet på ovan beskrivna sätt samt skydd mot att dokumentet ändras eller presenteras i sådan form eller i sådant sammanhang som är kränkande för upphovsmannens litterära eller konstnärliga anseende eller egenart.

För ytterligare information om Linköping University Electronic Press se förlagets hemsida http://www.ep.liu.se/

Copyright

The publishers will keep this document online on the Internet — or its possible replacement — for a period of 25 years from the date of publication barring exceptional circumstances.

The online availability of the document implies a permanent permission for anyone to read, to download, to print out single copies for his/her own use and to use it unchanged for any non-commercial research and educational purpose. Subsequent transfers of copyright cannot revoke this permission. All other uses of the document are conditional on the consent of the copyright owner. The publisher has taken technical and administrative measures to assure authenticity, security and accessibility.

According to intellectual property law the author has the right to be mentioned when his/her work is accessed as described above and to be protected against infringement.

For additional information about the Linköping University Electronic Press and its procedures for publication and for assurance of document integrity, please refer to its www home page: http://www.ep.liu.se/

© Johan Nilsson och Mikael Rothin