Construction of FPGA-based Test Bench for QAM Modulators

Master thesis performed in Electronic Devices
by
Josef Hederström
LiTH-ISY-EX--10/4381--SE
Linköping 2010-09-24
Construction of FPGA-based Test Bench for
QAM Modulators

Master thesis in Electronic Devices
at Linköping Institute of Technology
by
Josef Hederström

LiTH-ISY-EX--10/4381--SE

Supervisor : Professor Herbert Zirath
Dept. of Microtechnology and Nanoscience
at Chalmers University of Technology

Examiner : Professor Atila Alvandpour
Dept. of Electrical Engineering
at Linköping University

Linköping 2010-09-24
In today's fast evolving mobile communications the requirements of higher data rates are continuously increasing, pushing operators to upgrade the back haul to support these speeds. A cost effective way of doing this is by using microwave links between base stations, but as the requirements of data rates increase, the capacity of the microwave links must be increased.

This thesis was part of a funded research project with the objective of developing the next generation high speed microwave links for the E-band. In the research project there was a need for a testing system that was able to generate a series of test signals with selectable QAM modulations and adjustable properties to be able to measure and evaluate hardware within the research project. The developed system was designed in a digital domain using an FPGA platform from Altera, and had the ability of selecting several types of modulations and changing the properties of the output signals as requested. By using simulation in several steps and measurements of the complete system the functionality was verified and the system was delivered to the research project successfully. The developed system can be used to test several different modulators in other projects as well and is easily extended to provide further properties.
Abstract

In today's fast evolving mobile communications, the requirements of higher data rates are continuously increasing, pushing operators to upgrade the backhaul to support these speeds. A cost-effective way of doing this is by using microwave links between base stations, but as the requirements of data rates increase, the capacity of the microwave links must be increased. This thesis was part of a funded research project with the objective of developing the next generation high-speed microwave links for the E-band. In the research project, there was a need for a testing system that was able to generate a series of test signals with selectable QAM modulations and adjustable properties to be able to measure and evaluate hardware within the research project. The developed system was designed in a digital domain using an FPGA platform from Altera, and had the ability of selecting several types of modulations and changing the properties of the output signals as requested. By using simulation in several steps and measurements of the complete system, the functionality was verified, and the system was delivered to the research project successfully. The developed system can be used to test several different modulators in other projects as well and is easily extended to provide further properties.

Keywords: FPGA, Signal generation, Quadrature modulation, Software-defined radio, Testbench, Radio communication, High datarate.
Acknowledgements

I would like to give my thanks to Prof. Herbert Zirath for giving me the opportunity of developing a system within such a major research project, and to Simon He for being my supervisor and answering all the questions and helping me with problems that I could not resolve by myself. Without Simons careful explanations the task would have been very hard to resolve at some points. I would also like to extend my thanks to my examiner Atila Alvandpour for giving his support as involved in the research project previous to my thesis and giving me such freedom to examine the possibilities of the thesis. Finally, I want to thank my fiancé Anna-Karin for coping with the late nights at the office and in front of the computer.
# Contents

1 Introduction ........................................... 1
   1.1 Background .................................. 1
   1.2 MODEM Project ................................. 1
   1.3 Purpose ...................................... 2
   1.4 Technologies .................................... 3
      1.4.1 Cross Correlation ......................... 3
      1.4.2 Software defined radio .................... 4
      1.4.3 Modulation ................................ 4
         Basic modulation ............................ 5
         QAM Modulation .............................. 7
      1.4.4 I/Q Imbalance .............................. 8
         Amplitude imbalance ......................... 8
         Phase imbalance .............................. 9
   1.5 Structure ..................................... 10
   1.6 Limitations .................................... 10

2 Implementation ....................................... 11
   2.1 Tools ......................................... 11
      2.1.1 Programming language .................... 11
      2.1.2 Development environment ................ 11
      2.1.3 Hardware .................................. 12
      2.1.4 Measurement equipment .................. 14
      2.1.5 Project work flow ......................... 14
   2.2 PRBS ........................................ 15
      2.2.1 PRBS generator ............................ 15
      2.2.2 PRBS detector ............................. 16
   2.3 Modulators .................................... 19
      2.3.1 Conversion board limitations .......... 19
      2.3.2 QPSK modulator ........................... 20
         Matlab implementation .................... 20
         Verilog implementation ................... 20
         Modelsim simulation ....................... 21
      2.3.3 QAM modulators ............................ 22
         Quadrature Amplitude Modulation .......... 22
         8-QAM ....................................... 22
         N-QAM ....................................... 22
      2.3.4 System verification ....................... 24
   2.4 Measurement system .............................. 24
CONTENTS

2.4.1 Building blocks ........................................... 25
  Modulators .................................................. 25
  Power control unit ....................................... 26
  Control system ........................................... 26
  Predistortion ............................................. 28
2.4.2 Detailed overview ......................................... 28
2.4.3 System verification ....................................... 28

3 Results ......................................................... 30
  3.1 Setup .................................................... 30
    3.1.1 PRBS generator and detector ....................... 30
    3.1.2 Modulator ........................................... 30
    3.1.3 Measurement system ................................. 31
  3.2 Measurements ........................................... 34
    3.2.1 PRBS generator and detector ....................... 34
    3.2.2 Modulators ......................................... 34
      QPSK .................................................. 34
      8-QAM ................................................ 36
      16-QAM .............................................. 37
      32-QAM .............................................. 38
      64-QAM .............................................. 39
      128-QAM ............................................. 40
      256-QAM ............................................. 41
      1024-QAM ............................................ 42
    3.2.3 Measurement system ................................ 43
      Maximum output level ................................. 43
      QPSK ................................................ 46
      16-QAM .............................................. 47
      64-QAM .............................................. 48
      256-QAM ............................................. 49
      Predistortion ........................................ 50
  3.3 Results .................................................. 51
    3.3.1 Measurement issues ................................. 51
    3.3.2 Output power ...................................... 51
    3.3.3 Overshoots ........................................ 51
    3.3.4 Predistortion ..................................... 51
    3.3.5 Quality of measurement system ................... 53

4 Discussion .................................................... 54
  4.1 Thesis output ........................................... 54
  4.2 Future work ............................................. 54
    4.2.1 Predistortion ..................................... 55
    4.2.2 Phase Adjustment .................................. 55
  4.3 Insights ................................................ 55
  4.4 Final words ............................................. 56

Bibliography .................................................. 57
Chapter 1

Introduction

The work in this master thesis has been done in a research project at Chalmers University of Technology at the department of Microtechnology and Nanoscience, where the main task of the thesis had a focus on implementation of a test bench to be used within the research project.

1.1 Background

As the demand from customers of Internet capacity within mobile networks increase, the operators need to improve the capacity of the entire network continuously. To connect sub networks and antennas to the backbone of the network, several linking methods exist to create a backhaul. The research project in which the thesis work has been done, aims at developing a microwave link with increased data capacity to be used in a backhaul. Several participating institutions and industries are working together to develop this product from several different angles, to be able to present a state of the art product in 2013. The research project spans over several years and the thesis work is only a very small portion of the entire project.

The choice of the thesis is a result from the author’s interest in research and development of new technology, combined with the telecommunication profiled education at Linköping University. The author has also previously done project work in similar areas of expertise and is thus familiar with available technology and methods, which will reduce the dependency of guidance during the work progress.

1.2 MODEM Project

The thesis work is done within a joint research project between Chalmers, Linköping University, Ericsson AB, Gotmic AB, Bitsim AB, Sivers IMA AB and SP-devices. The purpose of the project is to demonstrate a 10 Gbps radio system over E-band and beyond in three years which would run during 2010 to 2013 with Chalmers Microwave Electronics Laboratory as the project driver. The following text is cited from the project specification [1] and provides a general idea of the purpose of the research project. A general knowledge in radio communications will ease the understanding.
This project is aiming at solutions for spectrum efficient radio communication at the E-band 71-76, and 81-86 GHz, and the 120 GHz band. The E-band is now gaining an increased interest for telecom operators providing internet access over the mobile network where radio links are used for the backhaul. The driving force is the demand from customers to have mobile access to internet for streaming video etc. The E-band provide 2*5 GHz bandwidth, commercial radio links for the E-band often use the modulation format OOK\(^1\), which is simple but not spectrum efficient. The goal of this project is to investigate solutions for spectrum efficient use of the E-band and higher frequencies like 120 and 220 GHz-band, and as a result increase the bitrate at least by a factor of four utilizing the same bandwidth. In this project, several solutions will be implemented and tested. The first solution is based on D-QPSK\(^2\) modulation, which uses a precoder realized in FPGA. This solution is more complex than the OOK-based system but is more spectrum efficient, by a factor of two. Like the OOK, the D-QPSK is non coherent i.e. the phase of the carrier is not needed to be retrieved. For higher spectrum efficiency, more complex modulation is required like QAM\(^3\). In a QAM signal, the data is represented by both amplitude and phase of the modulated carrier, the carrier phase reference have to be retrieved in the receiver, and both the amplitude and phase have to be detected. Therefore D/A converters are needed in the modulator of the transmitter and A/D converters in the receiver. Some of the simpler modulation formats will be realized in hardware while QAM will be realized using software in FPGA’s.

1.3 Purpose

The relation between the research project and the master thesis provides an understanding of the purpose. The project purpose is to investigate several different options and types of modulators and modulations, and the efficiency of these. To provide a flexible way of testing the transmission and modulation properties of products within the project, a proposal of developing a test bench as a master thesis was laid out. This thesis is the result of that proposal.

The focus within the master thesis was on implementation of a system to be used for testing products within the research project. The system that was implemented provides a way of measuring properties of different options that are selectable in the project to be a guidance in crossroads with focus on modulation type. Outside of the scope of the master thesis, the test bench will also be usable for other components in the project but this will not be considered during implementation due to time and resource limitations.

\(^1\)On-Off Keying, a very basic digital modulation

\(^2\)Differential Quadrature Phase Shift Keying, a slightly more complex modulation

\(^3\)Quadrature Amplitude Modulation, can be a very complex modulation
1.4 Technologies

Within the scope of the thesis, all work that was done was based on information of existing technology although the actual implementation was very specific for the project. The existing technology includes the types of modulation and techniques for generation of test sequences. References to these are presented within the respective implementation chapters. The following topics provide background information on some of the technologies and methods used within the thesis.

1.4.1 Cross Correlation

Detection of a bit sequence is a major part in the thesis, and one very sophisticated method of doing this is by cross correlation which is described in detail below. Performing cross correlation on a bit sequence is a widely used technique for detection of a transmitted sequence. The cross correlation can be used as a measure of similarity between two bit sequences as a function of time lag applied to one of the sequences. When performing cross correlation between a reference sequence and an input sequence, the correlation value will produce a peak when the two sequences match, e.g. the input sequence contains something similar to the reference sequence. The peak value in the digital domain corresponds to the matching number of bits between the two sequences. This peak can be used when detecting start of packets, or to verify the integrity of a signal that has been transmitted and received. A simple example is pictured in Figure 1.1 where two copies of the same sequence are cross correlated when a time shift has been applied to Sequence 2.

![Cross correlation example](image)

As seen in Figure 1.1, the resulting cross correlation has twice the length of the input sequences. This can be referred to by examining the convolution operation, since cross correlation is a convolution between the two sequences. The cross correlation is actually an operation that includes shifting one of the
signals and multiplying with the other to produce a sum which is a measure of how similar the signals are. The formula for cross correlation is the following:

\[(f \ast g)[n] = \sum_{m=-\infty}^{\infty} f^*[m]g[n + m]\]

where \(f^*\) denotes the complex conjugate of \(f\). To simplify the calculation, this operation is often done in the frequency domain where the operation can be performed as multiplying the complex conjugate of \(f\) with the Fourier transform of \(g\);

\[F\{(f \ast g)\} = (F\{f\})^* \ast F\{g\}\]

where \((F\{f\})^*\) denotes the complex conjugate of the Fourier transformed \(f\). By doing this operation and using an inverse Fourier transform on the result, a cross correlation between two signals has been made. Note that this implements a linear convolution and thus the input signals need to be zero padded to twice the length of the desired correlation sequence to avoid aliasing. To better understand this property, refer to the convolution operation in time domain whose result is twice the length of the operators.

1.4.2 Software defined radio

The general idea of constructing communication components in software by using FPGAs or MCUs has increased in popularity as prices decrease and performance increase. The technology is referred to as SDR, Software Defined Radio. This technology provides a way of changing properties of a radio transmitter or receiver within reconfigurable blocks, such as modulation, frequencies, amplitudes and algorithms amongst others.

SDR technology opens up new ways of adaptive communication and a higher spectral efficiency according to Tuttlebee [2]. Since the parameters can be controlled by software, a higher performance and greater spectral efficiency can be achieved since the communication channel can be adjusted dynamically depending on the environment, and properties such as modulation type can be adjusted to minimize the number of errors while keeping the data rate at a maximum. The interest in SDR continuously increases, but some problems exist that still require parts of the radio system to be in hardware and this is due to the limited performance of A/D and D/A converters and requirement of faster performance of the SDR [2], especially as frequencies increase. In the thesis, a SDR block was used as a main component in the test bench that was developed.

1.4.3 Modulation

The process of varying the properties of an high frequency analog signal, for example a cosine wave, is known as modulation. The signal that contains the properties that modulates the wave is known as the baseband signal and the high frequency signal is called the carrier signal. The purpose of a modulator is to change the properties of the carrier wave according to the baseband signal, resulting in a signal containing information that is suitable for transmission called the Radio Frequency (RF) signal. The baseband signal can either be a continuous wave with applied properties, an Intermediate Frequency (IF) wave,
or raw data as in the case with direct modulation which is explained later. Figure 1.2 illustrates a simplification of how the blocks are connected in a generic modulator to provide a better understanding. The purpose of this project was to provide a baseband signal that could be fed to a modulator for testing and measurement purposes.

![Figure 1.2: Generic modulator structure](image)

**Basic modulation**

The basic principle behind digital modulation is to modify the carrier signal according to the binary data input. The three fundamental methods for digital modulation which modifies the carrier signal in specific ways are

- Amplitude shift keying - ASK
- Frequency shift keying - FSK
- Phase shift keying - PSK

Each of the methods modulate the amplitude/frequency/phase of a carrier in a specified number of finite steps according to the input bits. These generally provide a transmission speed of 1 bit/s/Hz, but using them as building block for more complex modulation schemes provide means of increase higher speeds of transmission which is explained more in detail by J. Das in Review Of Digital Communication [3, page 194]. They each affect the carrier wave with different properties and thus the visual representation of the wave is changes, which is illustrated in Figure 1.3.

Another way to visualize the modulation techniques is by using a constellation diagram. Each of the diagrams in Figure 1.4 hold a different property on the x-axis.

In the different diagrams in Figure 1.4 only two separate binary values are represented and thus only two different values of the amplitude/frequency/phase are utilized. Depending on the modulation used, the points represent different properties being modulated. This corresponds to the ability to transmit either a '1' or a '0'. This can be extended by introducing an increased number of possible constellation points as in Figure 1.5 and as a result, a higher bit rate can be achieved since each point can represent several bits, known as codewords. The increased number of constellation points relate to the number of finite steps the amplitude/frequency/phase can be modulated as.

A received signal is seen as a point somewhere in the constellation diagram and the goal is to decode it as the correct transmitted point. The correct
Figure 1.3: Three different types of modulation on the same data

Figure 1.4: Constellation diagrams with different x-axis

Figure 1.5: Amplitude shift keying with four finite steps of amplitude, enabling four different constellation points
point must be estimated, since a transmitted signal is distorted by noise during transmission and thus the position of the point is distorted. The downside to an increased number of constellation points is the reduced SNR, *Signal to Noise Ratio*, when using the same transmission energy compared to a modulation with fewer constellation points as explained by J. Das [3]. The reason for this is when decoding, the received signal can be decoded into an increased number of points and thus it will be harder to make a correct decision on a signal that has been distorted by noise since the distance between the points decrease when using the same transmission energy, visualized in Figure 1.6.

![Figure 1.6: Different sized decision areas with four and two constellation points](image)

By using two separate modulation methods on a single carrier, it is possible to transmit more data on the same carrier using the same energy. The Figure 1.7 describes a transmission where the carrier has been modulated by both amplitude and phase and thus more codewords can be represented.

![Figure 1.7: Signal modulated by both ASK and PSK](image)

**QAM Modulation**

A commonly used modulation scheme is *QAM*, Quadrature amplitude modulation. According to Ergen [4, page 35], this is one of the most widely used modulation methods and is used in most of today's digital communications. As the name suggests, it is an implementation of amplitude modulation. By using two separate carriers phase shifted to each other the two carriers are amplitude modulated separately by different data, and then combined into a single carrier by simple addition. This results in a signal similar to the one in Figure 1.7 but has a simpler implementation due to the simplicity of amplitude modulation.

The two carriers are named I and Q, in-phase and quadrature, and are usually a cosine (I) and sine (Q) wave due to their 90° phase shifted relation to each
other. The finite number of amplitude steps that the I and Q can be adjusted to implements different types of QAM modulations. By having two selectable amplitudes for each carrier, a constellation diagram such as the leftmost in Figure 1.8 is obtained and as seen, where each point in the constellation diagram represents two bits of data. This comes from each of the two carriers having two selectable amplitude levels and can thus represent one bit each, and together two bits. If the number of amplitude steps are increased to four the data represented by each point is four bits. By increasing the number of selectable amplitude steps, and thus the number of constellation points, more data can be represented by each point and thus a higher transmission rate can be achieved and the loss of SNR. The number of constellation points dictates the modulation type, described as M-QAM where M is the number of points in the constellation.

![Figure 1.8: QPSK, 16-QAM and 256-QAM constellation diagrams](image)

There are several different implementations of QAM modulation such as differential, circular and rectangular. In this project only rectangular QAM was implemented due to time limitations and request from supervisors.

### 1.4.4 I/Q Imbalance

When transmitting a QAM signal as a radio wave, a modulator is used to apply the different amplitude levels on a carrier wave and thus modulating it. The method of directly applying amplitude levels on an input carrier wave is called direct conversion and due to the cost-effective implementation of this it is a common choice in today’s digital communication systems. However, according to Schenk and Linnartz in *RF imperfections in high-rate wireless systems* [5], it has a few disadvantages compared to other types of more conventional architectures of modulators and this thesis has a focus on one of these disadvantages: the I/Q imbalance.

This is caused by either phase or amplitude mismatches in the modulator that is supposed to modulate a carrier wave, causing transformations in the transmitted or received constellation diagram. There are also other sources of errors and types of errors that are not focused upon in this thesis, such as DC offset, LO leakage and Carrier suppression.

**Amplitude imbalance**

If either the I or Q signal has a higher conversion gain and thus higher amplitude steps than the other, the resulting constellation diagram will be squashed in
one direction. If this occurs during transmission, the receiving system will have severe problems with detection especially in the outer points when determining the correct position. It is of uttermost importance that a constellation diagram is symmetric as expected. An example of a squashed constellation diagram can be seen in Figure 1.9.

![Figure 1.9: IQ Amplitude imbalance](image)

![Figure 1.10: IQ Phase imbalance](image)

**Phase imbalance**

An I/Q modulator has an internal 90° phase shift from the input for the quadrature signal, resulting in two waves with a phase difference. This is one of the most critical components in an IQ modulator since it is the sole cause of phase imbalances within the modulator. If one of the signal paths is slight longer, or the phase shift is not a perfect 90°, the resulting signal will not have a symmetric shape but rather a diamond shape. A constellation diagram affected by a phase imbalance can be seen in Figure 1.10 with a clear diamond shape.
1.5 Structure

The thesis focus is around three main tasks that were decided at the start:

1. Generation of pseudo random data sequence
2. Generation of I/Q symbol signals for an E-band I/Q modulator for QPSK, 8-PSK, 16-QAM, 64-QAM, 128-QAM etc
3. Interfacing with existing modulator from Gotmic

The natural structure of these three tasks were to take them in the numbered order, since they each depend on the previous. Before the actual work could start, some time were spent setting up the development environment and getting familiar with the equipment to be used during the thesis.

The first two tasks were performed in the same way, both were preceded with gathering of information about different methods of implementation to select the most suitable for the project. A large amount of publications and papers were read to gather knowledge and possible benefits and drawbacks with the different methods. The major part would then be the implementation phase, with simulation and verification along with rewriting in several iterations until a satisfactory product had been produced.

The final task was to compose the building blocks previously implemented into a test bench to be used on existing hardware modulators. However due to lack of hardware resources, measurements upon a modulator could not be performed. Instead measurements on the final system was performed to evaluate the performance of it.

1.6 Limitations

At first the possible implementations were not known due to the hardware being used was relatively new. Therefore the limitations were open until tests had been done on the hardware after a few weeks of usage. In a perfect case the limitations would be chosen freely depending on needs but this was not the case since the hardware inflicted on the possible choices. The enforced limitations were decided in collaboration with the supervisors and were the following:

Rectangular constellations was focused on instead of the same modulations with different constellation point positions.

PRBS\(^4\) length 127 was decided to be used. Any length of \(2^n - 1\) could be chosen but \(n=7\) was after discussion determined to be enough for the project.

Demodulation would not be implemented, but instead focus would be on producing a well functional baseband signal generator that could be used and extended with demodulation outside the master thesis scope.

Amplitude correction was determined more important than Phase correction and was therefore prioritized.
Chapter 2
Implementation

The main focus of the thesis was in the implementation phase since the main objective of the thesis was to produce a hardware unit usable in the MODEM project.

2.1 Tools

To reach the goal of constructing a hardware unit with flexible attributes, the platform to be used in the project was chosen as an Field Programmable Gate Array (FPGA). This hardware component consists of logic elements with programmable paths in between, providing the possibility to create different hardware behavior by programming the FPGA. The programming is a result of code written in a hardware description language. In this project SystemVerilog was used by preference of the supervisor.

2.1.1 Programming language

The code used to describe the desired hardware behavior in an FPGA is written in a language type known as hardware description language. There are several different languages but they all produce the same product, a programmable file to set the programmable paths in an FPGA to modify the behavior as wanted. SystemVerilog, used in this project, is an extension to the widely used Verilog, providing additional data types and more flexible code construction than the original.

2.1.2 Development environment

The choice of environment when developing for FPGAs is tightly bound to the hardware used, since each hardware manufacturer provides the environment for their own product. In this project, the hardware was supplied from Altera and thus the most convenient development environment was Quartus II which has features especially useful for development on Altera hardware. Several advanced hardware units can easily be generated by the program, providing means of utilizing several advanced hardware features such as Phase-Locked Loops (PLL) and Fast Fourier Transforms (FFT).
To get a better understanding of the algorithms and methods to be implemented, simulations in Mathworks Matlab were performed prior to each implementation task. By breaking down and dividing complex operations into smaller steps, a better understanding of how an implementation could be done was gained. As an example, dividing the cross correlation operation into basic mathematical operations such as multiplication and addition provided a way of understanding the needed components. The smaller steps were then combined to perform the original task, and compared with built-in functions of Matlab for verification.

A major part of the development was simulation of the written code, as examining the programmed FPGA simply would not provide enough information of internal states and is also a very cumbersome task due to compilation and programming times. Since Quartus II has a good integration with Modelsim, also provided by Altera, the choice of simulation software was obvious. With the help of Modelsim, the expected behavior of the written hardware descriptive code could be examined by simulation.

### 2.1.3 Hardware

As an FPGA is only a chip, more hardware was necessary to utilize the chip. To load the compiled code into the FPGA, a programmer was needed and connections to the FPGA had to be established. Also input and output possibilities were required to make use of the system along with power supplies, reference clocks, buses and other peripherals. There were several boards equipped with these units, along with an FPGA, to make development easier.

The choice for this project was a board supplied from Altera, the *Altera DE3 development system* (Figure 2.1). The DE3 hosts many interesting peripheral interfaces but the ones focused on and needed by this project were the following:

- Stratix III FPGA.
- USB connection with on-board programmer.
- HSMC Extension board connectors.
- External SMA clock input and output.
- Switches and Pushbuttons.
- 8-Segment and LED display.

The FPGA runs at a speed specified by the PLLs inside the unit and the maximum clock frequency is dependent on the code programmed into the unit. As the DE3 does not host any analog output needed in the project, additional hardware was needed to provide an analog output. The DE3 has four HSMC ports for fitting daughter boards, therefore a dual analog to digital and digital to analog conversion board (Figure 2.2) was obtained for the analog output extension.

The conversion board had a maximum sample rate of 250 Msps (MSamples per Second) for digital to analog conversions, and a maximum rate of 150 Msps when using analog to digital conversion. The rates were given for each channel, and the separate channels also used separate differential clocks if necessary. The
Figure 2.1: Altera DE3 development board

Figure 2.2: Dual A/D and D/A conversion board
resolution for both conversions were 14 bits unsigned. The D/A chip on the conversion board had a differential output which was transformed into a single ended output on the conversion board, therefore the actual output voltage range was unknown and had to be measured. With a commonly used resistance of 50 Ω load on the digital to analog converter, the output ranged from 0-500 mV during initial measurements.

Since early implementations were focused on wide usage areas, the modulator was implemented in such a way that a demodulator could be used as well and thus the first implementations were limited to 150 Msps in both directions. As the work proceeded, two possible ways of implementation became clear and demodulation was no longer a focus.

**50 MHz IF** was the maximum that could be achieved if the analog output was to be a modulated IF wave that some modulators require. This is due to the number of sampling points needed to properly sample a combined sin and cos wave. Since the maximum sample rate of the D/A converter is 250 Msps, five points can be sampled each period and thus both the sine and cosine wave can be sampled.

**250 Msps direct modulation** could be used if the target hardware could handle the signal and was not in need of a IF signal. This puts stricter requirements on the hardware to be tested but also provides a much higher possible bit rate.

The two possible options were discussed in the MODEM project and a decision of focusing on a 250 Msps direct modulation signal was selected, but due to limitations in hardware the full speed of the conversion board could not be used without distortions in the conversion. Therefore, a sample rate of 200 Msps was selected. Also the option of selecting a 50 Msps rate was implemented to provide more stable output waveforms.

### 2.1.4 Measurement equipment

To observe the analog output from the conversion board, an Agilent Infiniium 54854A oscilloscope with X/Y-plot was used. This provided the ability to observe the constellation points produced in the analog domain, which is explained more in detail in the following sections.

### 2.1.5 Project work flow

The different tools specified all have their part in the work flow, and to illustrate the dependencies of each other and to visualize the work flow Figure 2.3 illustrates when the different tools were used. As the work flow describes, a hardware design was the starting point of the work. As soon as a design had been made, it was implemented in Matlab and Quartus II in two separate trails. The Matlab implementation resulted in immediate simulation and could be used find errors and provide a better understanding of the design early in the work flow. The Verilog code implemented in Quartus II resulted in HDL code that could be simulated using Modelsim, and by comparison with the Matlab simulation a first verification could be made before going deeper into implementation and synthesis of actual hardware.
When the Matlab simulation, Modelsim simulation and hardware synthesis all are completed, these can together perform a very genuine and reliable verification of the implemented hardware design.

2.2 PRBS

The first hardware unit to implement was a PseudoRandom Bit Sequence generator and a detector for such a sequence. A PRBS can be seen as a sequence of bits with a specific length where the bits appear to be random, and the purpose of a PRBS generator is to produce such a sequence. Due to the random appearance, two rotated copies of the same PRBS does not have many bits in common except for when the two sequences match up exactly. This is useful when examining the input and output relation of a unit under test which is described more in detail later. The goal of this first step was to produce a PRBS generator along with a detector to be used in test transmissions.

2.2.1 PRBS generator

The goal of a PRBS generator is to produce a sequence of a determined number of bits by using a generator polynomial implementation. There are several polynomials that can be used, and these polynomials generates sequences with the length

$$2^n - 1$$

where n is the degree of the polynomial. In this project, a decision was made that the length of 127 would be sufficient, thus the degree would be $n=7$. To implement a generator polynomial in hardware, $n$ registers are used in combination with XNOR gates to produce a linear feedback shift register (LFSR).

When put together, the degree of the polynomial decides the length of the bit sequence and the number of registers to be used. The polynomial provides information on where to place XNOR gates on the registers to create taps that implement a LFSR, whose output will be a PRBS. According to application note XAPP 052 [6, page 5] using XNOR from registers 6 and 7 will implement
a maximum length LFSR. This implementation is derived from the generator polynomial:

\[ y = x^7 + x^6 + 1 \]

The hardware implementation with the XNOR taps is described by the circuit diagram in Figure 2.4.

![Linear feedback shift register](image)

Figure 2.4: Linear feedback shift register

To implement this hardware in SystemVerilog, the following code was written:

```verilog
reg [1:7] lfsr;
reg lfsr_out;

lfsr_out <= lfsr[7];
```

In addition to this, control signals and interfaces along with additional signals fully implemented the PRBS generator as a hardware unit. To control this unit from other units, the interface was as specified in table 2.1.

<table>
<thead>
<tr>
<th>direction</th>
<th>type</th>
<th>name</th>
</tr>
</thead>
<tbody>
<tr>
<td>input</td>
<td>wire</td>
<td>clk</td>
</tr>
<tr>
<td>input</td>
<td>wire</td>
<td>reset</td>
</tr>
<tr>
<td>input</td>
<td>wire</td>
<td>enable</td>
</tr>
<tr>
<td>output</td>
<td>wire</td>
<td>out</td>
</tr>
</tbody>
</table>

Table 2.1: PRBS generator control signals

2.2.2 PRBS detector

To utilize the PRBS generator in a test transmission, a tool for detecting the reception of the signal was necessary and this is where the PRBS detector was used. The purpose of the detector was to find a PRBS within a chosen bit stream and provide a way of measuring the integrity of the found sequence. A reference signal to be located in the bit stream was needed during the detection and therefore the detector had a PRBS generator built in with the same properties as of the transmitting generators seen in Figure 2.5, these separate generators were not connected in any way.

As cross correlation is an effective method for comparison of signals and thus detection, the method was used in this implementation. To perform the Fourier transform, a fast Fourier transform (FFT, which is a more efficient way of calculating the transform) was used both for the transform and the inverse transform. The development environment Quartus II supplied tools for
construction of these units which was used since the construction of an FFT in a hardware description language such as Verilog is a rather complex task. The architecture of the implemented FFT and inverse FFT were generated as a streaming architecture, meaning that a new sequence can be fed into the FFT at the same rate all the time without any halts as seen in Figure 2.6, and the calculation takes exactly as long as it takes to feed a new sequence. This reduced the need for intermediate memories or buffers and simplified the design as the same bit rate was used throughout the detector.

The sequence to detect, the generated PRBS, was 127 bits long and thus the cross correlation sequences needed to be at least 254 bits long to avoid aliasing. As the FFT unit can only handle lengths in powers of two the length of the implemented FFT was 256.

Since the bits in the streams had values of either '1' or '0', both of the sequences had to be translated since, as mentioned earlier, zeroes do not affect the resulting value of the cross correlation. Therefore the first step in the PRBS detector was to convert the bits '1' and '0' in both sequences to the values '1' and '-1' represented as 8 bit floating point numbers since this was the simplest input the FFT units would accept. A possibility to simplify the design by implementing another FFT structure can be seen here but this was not in the scope of this project. Figure 2.7 illustrates a simplified example of the operation, and note that the boxes does not represent bits but rather values.

After transforming the two 256 bit sequences into the frequency domain, a multiplication between one sequence and the complex conjugate of the other was performed as a part of the cross correlation operation. The product of the multiplication was fed to the inverse FFT which completed the cross correlation operation. Each clock cycle, a new value was placed at the output of the inverse FFT which was the result of the operation and when a peak value was found, the two sequences match in time and values. Due to the implementation of several FFTs and multiplications, finding the location of the match in streaming
sequences is a timing issue. To simplify the interface and usage of the PRBS detector an internal buffer of 947 bits length has been implemented, along with a start of packet (SOP) signal. The signal timings can be seen in Figure 2.8.

The buffer was of type FIFO (first in, first out) and it buffered the input stream. When a peak was found by the cross correlation, the PRBS detector signaled this by setting the SOP signal high during one clock cycle, and at the same time the buffer output the first bit in the detected sequence. Thanks to this implementation, shown in Figure 2.9, a transmitted sequence can easily be located and observed by the PRBS detector. If a measurement of the number of correctly transmitted bits is required, the actual value of the cross correlation peak can be read as this is the definition of that value.

The signals used to control and utilize this unit are specified in table 2.2. Due to the delay in the FFT units the detector unit has a total delay of 947 clock cycles, but the system can be fed data every clock cycle due to the implemented streaming architecture. The complex nature of the FFT also reduces the highest possible frequency in the FPGA to around 420 MHz, this cannot be improved other than by changing FFT implementation. This was not an issue in this project due to other hardware limiting the maximum speed, but could be an issue in higher performance platforms.
2.3 Modulators

The main task of the project was to produce a baseband signal in the FPGA and, by using the attached conversion board, convert these digital signals into an analog set of signals to be fed into a modulator for testing. The signals to be fed to the modulator was the In-phase signal and the Quadrature signal. The QPSK modulator was one of the first components to be implemented and therefore the final design was revised when the unit was used in later stages. The main difference between this implementation and the one used later is the usage of an IF wave. This modulator was constructed to output I and Q signals as a 75 MHz modulated wave.

2.3.1 Conversion board limitations

To convert the generated digital signals into the desired analog signals, the conversion board was used. This board was also the main limiting factor in what frequencies and sample rates could be achieved due to conversion speed limits. The board features two A/D converters and two D/A converters which makes it a suitable candidate for the thesis work, and each of the four converters can be fed with a separate clock. The maximum sample rate for conversions from digital to analog is 250 Msps for each channel, and for analog to digital the limit is 150 Msps.

To properly sample the cosine and sine IF waves, the clocks to the quadrature channel on the conversion board was shifted by 90°. The sampling frequency was set to 150 Msps for possible usage of the demodulator channels, and therefore the frequency of the IF wave was set to 75 MHz to avoid folding of the signal according to the Nyquist limit which is explained more in detail by Cowpertwait.
Implementation and Metcalfe [7, page 181].

2.3.2 QPSK modulator

The building blocks of a Quadrature Phase Shift Keying modulator were quite straightforward. A data source was needed as well as a mapper that converted the phase data into constellation points. If an IF wave was to be used, a function generator to generate the sinus and cosine waves would be needed as well. To output the generated signal to the D/A conversion board, a differential clock was required by the board and this clock was generated by a PLL unit inside of the FPGA which also generated the system clock. Since the D/A board input was 14 bits unsigned, a conversion from signed to unsigned had to be done as well.

Matlab implementation

To get a better understanding of the inner workings of the modulator, a Matlab simulation was performed. In QPSK there are four constellation points, each representing a symbol of two bits, a codeword. A PRBS sequence was generated and fed into a mapper that would select the proper I and Q phases, 0° or 180° respectively depending on the data. As this was an early test and a final decision on system output had not yet been made, an IF wave at 50 MHz was used due to the hardware limitations explained earlier in "Tools". The two separate waves of I and Q as well as the combined wave were plotted and the result was observed.

As the result from the simulation matched the expected output, the basic structure in the system had been found. The simulation result would later be compared with simulation output from the constructed Verilog code.

Verilog implementation

The QPSK modulator was designed to run stand-alone, therefore all vital components were included in the implementation. The only inputs to the unit were a 50 MHz oscillator clock from the Altera DE3 board as well as pushbuttons and switches.

A PLL unit was created by the help of the Altera megawizard which enables usage of one of the hardware PLL units inside the FPGA. The input to the PLL was the 50 MHz on-board clock, and it outputs a clock of 300 MHz to be used internally as the system clock as well as sampling clocks to the conversion board of 150 MHz each with a 90° phase shift on one. To create the cosine and sine waves to be modulated, a Voltage Controlled Oscillator (VCO) was used as a function generator. As in the case with the PLL, it was generated by the Altera megawizard. It outputs two waves of 14 bits signed resolution each, a cosine and a sine. These were the two waves that was modulated by the data source, which was a PRBS generator.

Each of the two waves had two possible steps of modulation, in-phase or phase shifted 180°. The PRBS generator outputs a serial bit stream, and since a bit can take two possible values the mapping from bits to phase was rather straightforward. The symbol was changed each period and since each of the

1 Built-in tool of Quartus II to construct complex hardware blocks
waves needed one bit each the PRBS generator was programmed to output new bits in a rate of 150 bits per second. A multiplexer selected the wave to be modulated and this resulted in 75 bps to each of the waves. To align the arrival time of the two bits to modulate the waves, a delay of four clock cycles was introduced (300 Mhz / 4 = 75) on the I path. The two bits were passed onto the symbol mapping unit, which selected the value of the phase of the waves.

The phase of the waves were adjusted by multiplication by either 1 or -1 depending on the data mapped as this is essentially the same as a 0° or 180° phase shift. In the final stages, the cosine and sine waves were converted from 14 bits signed into 14 bits unsigned. Since the values of the signed waves oscillated between the full range of 14 bits signed numbers, -8192 to 8191, the unsigned conversion was made by adding 8192 to the amplitude of the waves resulting in an oscillation between 0 and 16384. The resulting two unsigned waves were sent to the conversion board via the Altera DE3 extension port to be converted into a pair of analog signals.

An illustration of the connections in the implemented systems can be seen in Figure 2.10.

![Figure 2.10: QPSK Modulator Verilog implementation](image)

**Modelsim simulation**

To verify the timings of the in Verilog constructed system, Modelsim was used. To test the constructed unit, a test bench had to be implemented to provide stimuli that normally would be produced by the hardware such as clock and inputs. The purpose of the simulation was to verify timings of the sampling clocks and values of output, as well as correct functionality of mapping. To aid in this verification, the previously simulated code in Matlab was used. Since the code written in Matlab as a PRBS generator functions exactly as the Verilog PRBS generator, the two separate simulations should in the ideal case produce the exact same waveforms. As seen in Figure 2.11 the two simulations behaves as expected with the same output signals, the phase shifts occurs in the same manner. The actual simulation had an extensive length but not all data is presented here.
2.3.3 QAM modulators

One of the goals of the thesis work was to produce a test bench system with the possibility to provide a baseband signal in several different modulations. With QPSK being completed, the remaining modulations were extensions to this completed design. In these higher order modulations, the data source as a PRBS was not a desired component since there was no guarantee that all constellation points would be mapped or evenly distributed. Instead, a counter was used to make sure all possible constellation points were transmitted. In all of the following test systems, the data source counts up through all the possible constellation points.

Quadrature Amplitude Modulation

The previous design of QPSK utilized a phase shift of the waves, but this could also be seen as a amplitude shift between the steps of 1 and -1 since they affect the wave in the same way. QAM affects the amplitude of the two waves and modulates these into a finite number of amplitude levels. Since the same maximum output energy was used in all modulations, the number of levels were increased while utilizing the same bit range, 14 bits unsigned, reducing the distance between constellation points as the number of levels increase. A simple way to increase the steps between -1 and 1 would be to use smaller steps with decimals, but working with decimal point numbers in a digital domain is a very complex task. Instead a reduction of the amplitude on the sine and cosine waves were used along with increased values of multiplication when adjusting the amplitude of the waves.

8-QAM

This modulation increased the number of constellation points from the four in QPSK to eight. Since the same bit range is used, 14 bits unsigned, the sine and cosine wave amplitudes to be modulated were reduced by one bit and the symbols that would perform the modulation were increased by one bit. The constellation diagram of an 8-QAM transmission is shown in Figure 2.12. The additional constellation points are located on the level '0', therefore no increment of amplitude levels had to be done but this was observed in a later stage after the implementation was finished. The possible levels of amplitude were adjusted to -2, 0 or 2 but this could have been implemented by using -1, 0 and 1 although the resulting output is unaffected by this design.

N-QAM

As the number of levels of amplitude were increased in the following modulations, an even higher bit range for the symbols were needed. This resulted in a further reduction of the amplitude on the sine and cosine waves but still the same maximum output energy was used. As some multiplications resulted in utilizing the full possible range of the 14 unsigned bits and some did not, due to integer multiplication, the exact maximum output energy was not the same in all modulations. As the modulation order increased up to 1024-QAM the number of constellation points increased along with the levels of amplitude, and therefore the amplitude of the initial sine and cosine waves had to be reduced.
Figure 2.11: Modelsim and Matlab simulations comparison

Figure 2.12: 8-QAM Constellation diagram
The resulting properties of all the implemented modulators can be seen in Table 2.3.

<table>
<thead>
<tr>
<th>Modulation</th>
<th>Amplitude levels</th>
<th>VCO resolution bits</th>
</tr>
</thead>
<tbody>
<tr>
<td>8-QAM</td>
<td>0, ±2</td>
<td>13</td>
</tr>
<tr>
<td>16-QAM</td>
<td>±1, ±3</td>
<td>12</td>
</tr>
<tr>
<td>32-QAM</td>
<td>±1, ±3, ±5</td>
<td>11</td>
</tr>
<tr>
<td>64-QAM</td>
<td>±1, ±3, ±5, ±7</td>
<td>11</td>
</tr>
<tr>
<td>128-QAM</td>
<td>±{1, 3, 5, 7, 11}</td>
<td>10</td>
</tr>
<tr>
<td>256-QAM</td>
<td>±{1, 3, 5, 7, 11, 13, 15}</td>
<td>10</td>
</tr>
<tr>
<td>1024-QAM</td>
<td>±{1, 3, 5, ..., 27, 29, 31}</td>
<td>9</td>
</tr>
</tbody>
</table>

Table 2.3: Properties of higher order modulators

2.3.4 System verification

The final verification of the modulators was done by observing output analog waveforms and constellation points using an oscilloscope. To verify that the output was correct, the amplitude levels of the waves were observed as well as the phase shifts caused by the modulations and the difference in phase between the two waves I and Q.

As construction and simulation in Matlab is a rather time consuming task, the higher order modulations were verified by only observing the output waveform and constellation diagrams on an oscilloscope since the main objective of development of these modulators was to obtain building blocks for further implementations. The resulting output can be seen in chapter 3.

2.4 Measurement system

The purpose of the measurement system was to provide a hardware unit with selectable modulations and signal properties to provide a way of measuring the performance of different modulations and the I/Q imbalance of these in a modulator. This was to be measured and be of assistance when deciding the modulation type to focus on in the research project. The different modulation units that were implemented previously in the thesis were used as building blocks for this system, but they were all modified to suit the purpose better. It is important to note that the hardware target to be measured upon is a modulator constructed within the MODEM-project, and internally inside the FPGA several modulators are implemented to produce the test signals for the hardware modulator. These are two separate items which are not to be confused with each other although they are referred to by the same name. Since not all modulations were equally interesting from the research project point of view, a decision was made to only implement QPSK, 16-QAM, 64-QAM, and 256-QAM. Along with this, an ability to adjust the voltage of the I and Q signals individually was also needed. An overview of the system and the environment is pictured in Figure 2.13.

All of these functionalities had to be selectable and adjustable while the system was running, and the entire system also needed to be independent in
a way that no computer would be needed during testing. Since an FPGA is a volatile device\(^2\) the system had to be reconfigured at each power-up. To avoid the need of a computer to do this, an on board flash memory served as the source for programming the FPGA upon startup.

2.4.1 Building blocks

The measurement system was constructed by using several building blocks previously written, along with a power control unit and a control system. The following sections describe the inner workings of these.

Modulators

The system was built to test direct modulators\(^3\) and thus there was no need for an IF signal to be produced since only the amplitude levels of the I and Q signals were needed. The direct modulators internally produces a phase shift to be used on the Q signal from an input IF signal. Because of this, the VCO that resided in each of the modulators could be removed and instead the multiplication that previously was made with a sine or cosine wave could now be performed by multiplication with a constant factor. This constant was selected differently in each of the modulators to utilize the full range of the D/A converter and are described in table 2.4. The difference between the modulations were the number of amplitude steps within the same output range.

Since the signals from all different modulators later were to be adjusted in voltage, the conversion into unsigned signals had to be moved outside of the modulators and into the power control unit. Therefore the output was changed from 14 bits unsigned info 14 bits signed within each of the modulators.

To avoid several clock domains, the PLL that each modulator was equipped with internally was also removed and instead the clock signal was fed into the

---

\(^2\)Configuration is lost when power is turned off  
\(^3\)A direct modulator converts directly from data to RF frequency without the need of an IF signal
modulator from a PLL that was global for the entire measurement system. The clocks for the D/A conversion board was also moved in the same way, reducing the complexity of each modulator unit.

**Power control unit**

A requested functionality was to be able to adjust the output voltage of the I and Q signals individually while the system was running. Since the available signals utilize the full range of the D/A conversion board, the voltage of the output signals could not be increased but only decreased, and this was performed by a power control unit that would attenuate the signals within a certain range.

Since the signals should be scaled proportionally on each amplitude levels, a division with a voltage level variable had to be done. The desired division variable would be in the range of 0.5-1.0, but since division with decimal point numbers is a rather complex operation for an FPGA this was implemented using a different approach.

By performing a multiplication with \(2^{14}/x\) where \(x\) is the desired division variable and taking the high 14 bits of the 28 bit result, a much more effective division has been made since multiplication between two numbers is a much simpler task than division for an FPGA.

The maximum output from the D/A Conversion board was around 500 mV as measured previously, and without any attenuation the generated signals can at some stage reach this level. Eight stages of voltage levels were implemented since this was easily represented by the eight LEDs on the Altera DE3 board, and an estimate of the attenuation levels of these can be seen in table 2.5. As seen in table 2.5, the steps of maximum output amplitude are adjusted in steps of 15 mV.

Another functionality of this unit was to convert the resulting signed 14 bit signals into unsigned 14 bits since this unit was removed from the modulators and the D/A Conversion board does not handle signed signals. This was simply done by adding \(2^{14}/2\) to the resulting signal, changing the range from -8192 – 8192 into 0 – 16383.

**Control system**

As this system was to be used stand-alone, the need for a computer had to be eliminated. The programmed FPGA looses all information once it is turned off and therefore the program had to be stored on a flash memory on the Altera DE3 which could reprogram the device every time it was powered up. To control the system and get information about applied settings, the on-board switches, pushbuttons, LEDs and HEX display was used. Using the switches, different modulations could be selected and displayed on the HEX display, see table 2.6.

Pushbuttons were used to adjust the output voltage in eight stages, and to display the current setting the on-board LEDs were used. These were RGB LEDs therefore it was natural to use two separate colors for the different channels: Blue for I-voltage and red for Q-voltage. These were adjusted individually by using the four on-board pushbuttons as seen in Figure 2.14.
### Table 2.4: Multiplication constants and range

<table>
<thead>
<tr>
<th>Modulation</th>
<th>Amplitude range</th>
<th>Constant</th>
<th>Output range</th>
</tr>
</thead>
<tbody>
<tr>
<td>QPSK</td>
<td>-1...1</td>
<td>8190</td>
<td>-8190...8190</td>
</tr>
<tr>
<td>16-QAM</td>
<td>-3...3</td>
<td>2730</td>
<td>-8190...8190</td>
</tr>
<tr>
<td>64-QAM</td>
<td>-7...7</td>
<td>1170</td>
<td>-8190...8190</td>
</tr>
<tr>
<td>256-QAM</td>
<td>-15...15</td>
<td>546</td>
<td>-8190...8190</td>
</tr>
</tbody>
</table>

### Table 2.5: Voltage level configurations

<table>
<thead>
<tr>
<th>Voltage level</th>
<th>Attenuation</th>
<th>Division variable</th>
<th>Maximum output</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0.78</td>
<td>12800</td>
<td>390 mV</td>
</tr>
<tr>
<td>1</td>
<td>0.81</td>
<td>13312</td>
<td>405 mV</td>
</tr>
<tr>
<td>2</td>
<td>0.84</td>
<td>13824</td>
<td>420 mV</td>
</tr>
<tr>
<td>3</td>
<td>0.87</td>
<td>14336</td>
<td>435 mV</td>
</tr>
<tr>
<td>4</td>
<td>0.90</td>
<td>14848</td>
<td>450 mV</td>
</tr>
<tr>
<td>5</td>
<td>0.94</td>
<td>15360</td>
<td>470 mV</td>
</tr>
<tr>
<td>6</td>
<td>0.97</td>
<td>15872</td>
<td>485 mV</td>
</tr>
<tr>
<td>7</td>
<td>1.00</td>
<td>16383</td>
<td>500 mV</td>
</tr>
</tbody>
</table>

### Table 2.6: Switches configuration

<table>
<thead>
<tr>
<th>Switch1</th>
<th>Switch0</th>
<th>Modulation</th>
<th>HEX Display</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>QPSK</td>
<td>4</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>16-QAM</td>
<td>16</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>64-QAM</td>
<td>64</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>256-QAM</td>
<td>56</td>
</tr>
</tbody>
</table>

Figure 2.14: Voltage adjustment controls on Altera DE3 board
Predistortion

Predistortion was considered during the late stages of the thesis as it would be of great assistance to be able to adjust the individual points in a constellation diagram using a computer. Using a hardware block containing user input regarding individual constellation point properties, it could be used to adjust the constellation diagram prior to sending the signals to the output and thus countering the effects of phase and amplitude mismatches. This is a more comprehensive task since the communication with a workstation must be implemented and also makes the system less stand-alone. Right now the system can operate fully without and peripherals but a workstation is needed since adjusting points individually with a matrix require a better input and visualization of data.

As a trial, a proof of concept with predistortion was implemented. It provided the possibility of adjusting two points in the 64-QAM constellation diagram by a switch. This implementation did not provide any useful functionality, other than proving that adjustment of individual points could be made. In further development of the measurement system, this could be useful and it is discussed more in detail in chapter 3.

2.4.2 Detailed overview

All of the described building blocks are connected together as described in Figure 2.15. The data source fed all the modulators the same data, and each modulator mapped the data into symbols of I and Q. This data was fed into two multiplexers, selecting the proper modulation, and fed it forward to the power control unit which attenuated the signal according to the user. The conversion from signed into unsigned was also performed by the power control unit, and finally the signal was fed to the two D/A Converters on the conversion board. The predistortion block is not pictured due to it being only a proof of concept, but it was implemented inside of the 64-QAM modulator.

![Figure 2.15: Measurement system detailed overview](image)

2.4.3 System verification

The measurement system consists of several building block that previously had been verified, although they have been modified afterwards to construct this
unit. To verify that the measurement system behaved as expected, an oscilloscope was used to observe the signal outputs and constellation diagrams with the different power levels and modulations. All measurement and results can be seen in the later Results chapter.
Chapter 3
Results

All of the implemented systems were measured and verified. The results of these along with comments and discussions are provided in this chapter.

3.1 Setup

Each of the hardware units implemented required different types of measurements for verification and thus different setup.

3.1.1 PRBS generator and detector

Since the PRBS units did not provide any output to be measured upon, all measurements were made by observing the internal states of the FPGA and comparing the output with simulations from Modelsim and Matlab. These produced enough information to prove the functionality of the system.

To verify the design, observations of the internal states and registers were needed and for this task, a built-in program in Quartus II known as SignalTap was used. This program had the ability to observe a selected number of registers on the actual FPGA during runtime and download them to the development environment for verification. This method was used to observe the value of the cross correlation during runtime when correct or distorted PRBS were sent to the detector.

3.1.2 Modulator

All modulator implementations were tested during development to verify the functionality and signal behavior. The output waveforms were observed by using an Agilent Infinium 54854A oscilloscope and measured upon to determine the maximum output voltages. The first implementation, the QPSK modulator, was also compared with a Matlab simulation during development to provide knowledge on how the output waveforms should appear if fully functional. The following implementations did not include this step as the expected waveform simply had an increase in the number of attainable amplitude levels as seen in the oscilloscope measurements. The only reason the Matlab simulation was used was to determine that the mathematical operation of the modulator matched the hardware, and the same operation was used in all later implementations.
Since each of the modulator implementations was a complete hardware block, the Altera DE3 had to be reprogrammed between each of the verifications. The waveforms were obtained using a 1 Gsps sample rate in the oscilloscope to properly sample the waveforms. The usage of SignalTap software was not possible due to hardware limitations when using the conversion board, therefore observations using an oscilloscope was the method of verification.

To obtain an eye diagram, the oscilloscope was set to keep the sampled waveforms for 500 ms while sampling new waveforms. This results in several samples being presented on the screen on top of each other, providing a possibility to see the amplitude levels at the same time in the shape of an eye, as seen in Figure 3.1.

![Figure 3.1: Eye diagram of I and Q signal](image)

### 3.1.3 Measurement system

To prove functionality of the measurement system and determine the actual values of the generated signals, several measurements were made with the fully functional final system. After measurements were performed, no modifications were made to the system.

The interesting data to acquire was information about:

- Maximum and minimum output level
- Step size of amplitude adjustments
- Distance between constellation points
- Functionality of the entire system
- Possible distortions or noise issues

All of the data was obtained using an Agilent infiniium 54854A oscilloscope, with a bandwidth of 4 GHz and a maximum sample rate of 20 GSps. The oscilloscope was connected in with 2 channels in X-Y mode with the measurement system,
in the Microwave Electronics Lab (MEL), at Chalmers MC2. Since the system had the configuration stored on-board, no programming with a computer was required after booting up the Altera DE3.

During the measurements, the symbol rate of the measurement system was set to 50 Msps in all cases. The system was therefore running at the same speed, 50 MHz, due to the implementation of the system. The oscilloscope was connected at inputs 1 and 2 which had a 50 Ω load with a BNC connector on the oscilloscope end and SMA connector in the other end from the conversion board D/A outputs. The conversion board was attached to the Altera DE3 with the measurement system programmed into the FPGA. The lab setup is pictured in Figure 3.2.

![Figure 3.2: Lab setup](image)

To adjust the properties of the output signal, the four on-board switches and four pushbuttons were used as explained in table 3.1

<table>
<thead>
<tr>
<th>Component</th>
<th>Adjustment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Switch 3</td>
<td>Predistortion enable</td>
</tr>
<tr>
<td>Switch 2</td>
<td>Sample rate selection</td>
</tr>
<tr>
<td>Switch 1</td>
<td>Modulation type</td>
</tr>
<tr>
<td>Switch 0</td>
<td>Modulation type</td>
</tr>
<tr>
<td>Button 3</td>
<td>Decrease Q</td>
</tr>
<tr>
<td>Button 2</td>
<td>Increase Q</td>
</tr>
<tr>
<td>Button 1</td>
<td>Decrease I</td>
</tr>
<tr>
<td>Button 0</td>
<td>Increase I</td>
</tr>
</tbody>
</table>

Table 3.1: On-board controls and properties

As a trigger for the oscilloscope sampling, the sampling clock from the D/A conversion board was used. Every time the digital signal to the conversion board
changed, the analog output signal had some overshoot before settling to the final new level. Since the sampling and thus the analog output changed at every rising edge of the sampling clock, caution was taken not to sample directly after this change since this would cause measurements to be taken when overshoots would be at its largest. See Figure 3.3 for a theoretic example of an overshoot.

Figure 3.3: Overshoot example

The sample rate of the oscilloscope was set to the same as the sample rate of the conversion board, 50 Msps, resulting in one sample per symbol. Using higher sample rates it was possible to observe overshoots being sampled so in a measure to avoid this the lower sample rate was used in the oscilloscope.

The measurements of voltages between points and levels were made by using internal markers in the oscilloscope and the data was stored by saving the screen output to a USB memory. All measurement results can be found in the following section. To plot the two signals as a constellation diagram, the 'versus' mode in the oscilloscope was used, plotting channel 1 versus channel 2 resulting in a X-Y plot as a constellation diagram.
3.2 Measurements

3.2.1 PRBS generator and detector

This verification showed that the implementation was fully working, as can be seen in Figure 3.4 where a peak of value 128 has been found. This data was obtained by using SignalTap logic analyzer, which captures data from the running FPGA and presents it to the user on a workstation while running.

data_in is an input stream of data, and when this data was sampled the input stream was fed an PRBS sequence that matched the expected sequence two times in a row. The second sequence was also detected as can be seen 127 clock cycles later in the Figure. The buffered output data seen at data_out matches the sequence start when data_sop goes high, as expected.

![Figure 3.4: SignalTap internal observations of PRBS detector](image)

This hardware unit was declared as fully functional at this stage and no further development of it was performed. It was used as a building block in some of the work followed, but was also removed in the final stage of development of the measurement system since it was declared as unnecessary.

3.2.2 Modulators

All of the implemented modulators were verified by observing the eye diagram of the signal and measuring of the output voltage levels. The maximum output voltage was expected to differ due to the integer multiplication explained in chapter 2, Implementation. In all oscillator screenshots provided below, the yellow signal is the I signal and the green refers to the Q signal.

QPSK

Since this was the first modulator to be implemented, initial simulations with Matlab and Modelsim was done to perform a verification of the output waveform correctness. As seen in Figure 3.5 the two different simulation results produce the same output, although the actual phase shift occurs slightly later in the Matlab simulation. This does not impact on the output waveform correctness.

When programming the simulated code onto the Altera DE3 platform and attaching the conversion board output to the oscilloscope, the waveform pictured in Figure 3.6 was observed. The internal function generator produced a sinus and cosine wave of exactly 14 bits, therefore the output is at its maximum of 500 mV as measured by the two horizontal markers on the I signal.
Figure 3.5: QPSK simulations in Modelsim and Matlab

Figure 3.6: QPSK eye diagram
8-QAM

This modulation increased the number of representable amplitude levels by one, adding the zero level. This is clearly observed in the output waveform as the signal also passes at 0 V level, Figure 3.7. The maximum output level is still the same as it was possible to perform a multiplication that utilizes the full range of 500 mV peak-to-peak.

Figure 3.7: 8-QAM eye diagram
16-QAM

As the number of levels increased to four, the values of multiplication of the sinus and cosine waves could not utilize the full range of the conversion board 14 bits. It can be observed in Figure 3.8 that the output level has been reduced due to this. The four levels for multiplication used were -3, -1, 1 and 3. Multiplying the 12 bit generated waves with the odd number 3 cannot fill the entire range of the 14 bits output, since $2^{12} \times 3 = 12288$ and $2^{13} \times 3 = 24576$ which is outside of the range. That is why the maximum output value can only reach 364 mV, which is reasonable according to the maximum value:

$$500mV/2^{14} \times 2^{12} \times 3 = 375mV$$

![Figure 3.8: 16-QAM eye diagram](image-url)
32-QAM

Increasing the number of amplitude levels to six further impacted on the maximum output value due to the same issue as in the 16-QAM case explained above. Since the resolution of the internal sinus and cosine waves was reduced to 11 bits to fit within the output range and the maximum multiplication value was 5, the maximum output level value now was expected to be around

$$500\text{mv} / 2^{14} \times 2^{11} \times 5 = 312.5\text{mV}$$

The Figure 3.9 proves that this is the case, although a slightly higher value was measured due to limited measurement precision in the oscilloscope.

Figure 3.9: 32-QAM eye diagram
64-QAM

As this modulation increased the maximum amplitude multiplication value while remaining within the range of the output, the maximum output value was increased much closer to the limit. The number of amplitude steps were increased to eight and the theoretical maximum output value was

\[ 500\text{mV}/2^{14} \times 2^{11} \times 7 = 437.5\text{mV} \]

which is very close to the measured value in Figure 3.10.
128-QAM

This modulator suffered from distortions in the I channel which can be seen in Figure 3.11. The cause of these distortions could not be found but was thought of being results from overshoots due to large output voltage swings that the conversion board could not fully perform on par to. Due to this, the modulator was not selected in the further development in the thesis as this issue could not be resolved. With the twelve output levels, the expected maximum output voltage of the channels was expected to be

$$\frac{500\text{mV}}{2^{14}} \times 2^{10} \times 11 = 343.75\text{mV}$$

and this further proves that the multiplication was not performed as expected, as the measured maximum output voltage on channel I was measured to 392 mV.

![Figure 3.11: 128-QAM eye diagram](image)
256-QAM

Increasing the number of amplitude levels to 32, this modulation puts high demands on the conversion board being able to provide a good enough resolution between the levels. Observing Figure 3.12 it is hard to tell how well the resolution is due to the actual resolution of the screen on the oscilloscope. Due to this a zoomed in version is presented in Figure 3.13 where it is possible to see that the separation between the amplitude levels is still noticeable. The utilization of the range on the conversion board was almost used at maximum in this modulator, using $15$ as the maximum multiplication value and having a resolution of 10 bits on the sinus and cosine waves the expected maximum output voltage was

$$500mV/2^{14} \times 2^{10} \times 15 = 468.75mV$$

which can be compared to the measured level of 472 mV in Figure 3.13.

Figure 3.12: 256-QAM eye diagram

![Figure 3.12: 256-QAM eye diagram](image)

Figure 3.13: 256-QAM eye diagram zoomed in

![Figure 3.13: 256-QAM eye diagram zoomed in](image)
1024-QAM

Using 64 amplitude steps within a resolution of 500 mV, this implementation served as a performance evaluation of the conversion board rather than a block to be used in further implementations. As 31 was the maximum value of symbol multiplication, this implementation was very close to the limit of the output;

$$500\text{mV}/2^{14} \times 2^9 \times 31 = 484.375\text{mV}$$

As seen in Figure 3.14 this is close to what was expected. To properly see the separation between the amplitude levels, a zoomed in version of the I signal is seen in Figure 3.15. The separation can clearly be seen, but it can also be observed that the variation within the levels are actually greater than the distances between the different levels which most likely will cause problems when using this modulation together with a modulator or amplifier when noise is introduced to the signal. Therefore this modulation is not desirable.

Figure 3.14: 1024-QAM eye diagram

Figure 3.15: 1024-QAM eye diagram zoomed in
3.2.3 Measurement system

The following modulation types were implemented in the measurement system and measured upon; QPSK, 16-QAM, 64-QAM and 256-QAM. The measurements made were of maximum output levels and the adjustable range of the constellation point distances, since these were numbers of importance to the MODEM-project. Also the predistortion proof of concept was measured upon afterwards. The results are described in the following sections.

Maximum output level

As the same range was used in all of the implemented modulators, the exact same maximum output value was expected in all cases. The Figures 3.16, 3.17, 3.18 and 3.19 prove that the same maximum out value was measured in all cases. However the expected 500 mV output was only reached during overshoots as seen in Figure 3.20.

Figure 3.16: QPSK maximum output level
Figure 3.17: 16-QAM maximum output level

Figure 3.18: 64-QAM maximum output level
Figure 3.19: 256-QAM maximum output level

Figure 3.20: Maximum output level during overshoot
QPSK
Since this constellation only consists of a total of four points, the distance between points is the same as the maximum and minimum amplitudes. The Figures 3.21 and 3.22 provides information about the maximum and minimum output levels:

- The maximum distance between points was 372 mV
- The minimum distance between points was 291 mV
- Each point could be moved at a resolution of $(372 - 291)/7 = 11.5 mV$

Figure 3.21: QPSK maximum distance between points

Figure 3.22: QPSK minimum distance between points
16-QAM

This constellation consists of 16 points, each theoretically representing 4 bits of data. The Figures 3.23 and 3.24 show the measurement results.

- The maximum distance between points was 126 mV
- The minimum distance between points was 94 mV
- Each point could be moved at a resolution of \( (126 - 94)/7 = 4.5 mV \)

![Figure 3.23: 16-QAM maximum distance between points](image1)

![Figure 3.24: 16-QAM minimum distance between points](image2)
64-QAM

This constellation consists of 64 points, each theoretically representing 6 bits of data. The results were measured in Figure 3.25 and 3.26 and are:

- The maximum distance between points was 55 mV
- The minimum distance between points was 41 mV
- Each point could be moved at a resolution of $(55 - 41)/7 = 2mV$

Figure 3.25: 64-QAM maximum distance between points

Figure 3.26: 64-QAM minimum distance between points
256-QAM
This constellation consists of 256 points, each theoretically representing 8 bits of data. The results were measured in Figure 3.27 and 3.28 and are:

- The maximum distance between points was 25 mV
- The minimum distance between points was 19 mV
- Each point could be moved at a resolution of \((25 - 19)/7 = 0.86\text{mV}\)

Figure 3.27: 256-QAM maximum distance between points

Figure 3.28: 256-QAM minimum distance between points
Predistortion

As this was only a proof of concept, no valuable measurements were made but rather a proof of the possibility to move individual constellation points. In this example, two points were moved in both I and Q amplitude closer to the center. The Figures 3.29 and 3.30 proves that two points can individually be adjusted without affecting any of the others, and this has been done in a 26 mV step in both I and Q amplitude for each of the points.

Figure 3.29: Lab setup

Figure 3.30: Lab setup
3.3 Results

3.3.1 Measurement issues

In most of the measurements it can be observed that the Y-distance, the quadrature amplitude, between points is slightly higher than the X-distance. This can be caused by either attenuation in the connections, attenuation in a cable or by the D/A converters not having the exact same properties, since the digital value fed to the channels are in the exact same range and has the same values. On the other hand, this error is easily corrected by the amplitude adjustment possibility in the measurement system. After all measurements were completed, additional measurements were made with different set of cables and this proved that the attenuation was cause by the cables used during the measurements. As seen in the Figures 3.31 and 3.32 both measured on the X-Y plot of the QPSK signal, the cables was the sole cause of measurement variations in the amplitudes. The markers measure a difference in the maximum amplitude between the two channels:

- Cables used during measurement produced 501 mV and 514 mV maximum output on the channels.
- Replacement cables produced 494 mV on both channels.

3.3.2 Output power

The maximum output voltage was measured to 500 mV at a 50 Ohm load which represents the maximum range of the 14-bit digital values supplied to the conversion board, the distance between the digital values 0 and 16383. The D/A converter chip is specified at a higher output voltage, but due to conversion from a differential output voltage of the D/A converter to a single ended on the conversion board, the voltage dynamic range is reduced to the measured. It can be seen that the maximum output power could reach 500 mV at most during overshoots, and the actual maximum amplitude level was slightly lower.

3.3.3 Overshoots

None of the above measurements contain overshoots, this is thanks to good sampling and trigger points. However, in earlier measurements the overshoot was observed and as an example the Figure 3.33 has been attached which was taken with a high sampling rate, causing both the overshoot and the actual value to be sampled. This is an example of what it might look like when sampling is not done with caution.

3.3.4 Predistortion

It is clear that the 64-QAM signal has been adjusted by a predistortion setting on two individual points. Since their respective amplitude has been adjusted in both I and Q voltage at the same time, this has moved them slightly towards the center of the constellation diagram. In this project and this proof of concept, this does not provide any desired functionality but is rather a proof that any point can be individually moved without affecting any of the other properties. There is a possibility to extend this functionality and to adjust these properties using a computer but this is not within the scope of this thesis.
Figure 3.31: Measurement of overshoots made with the same cables as in the measurements

Figure 3.32: Measurement of overshoots made with a new set of cables
3.3.5 Quality of measurement system

By observing all of the above measurements, it can clearly be seen that the system does not contain any major distortions or miscalculations. The constellation points are perfectly symmetrical in the diagram both as full power and when attenuated by multiplications. The power adjustment worked as expected and all eight levels of power could be utilized in all four types of modulations.

The system can be considered very stable and fully functioning with all the expected properties, and can easily be extended with the functionality of predistortion.
Chapter 4

Discussion

After verification of the measurement system it is clear that the system performs to expectations. The output from the system are a pair of I/Q signals that can be directly applied to a modulator to evaluate the properties of the unit, as required.

4.1 Thesis output

The work was set out to produce the following three tasks;

1. Generation of pseudo random data sequence
2. Generation of I/Q symbol signals for an E-band I/Q modulator for QPSK, 8-PSK, 16-QAM, 64-QAM, 128-QAM etc
3. Interfacing with existing modulator from Gotmic

As the thesis has been completed, the work has lived up to the requirements. The pseudo random data sequence was implemented first of all, with a detector accompanying it. All of the required modulators and several more have been implemented in a complete dynamic system to be used with any direct modulator. Interfacing with the existing modulator could not be performed due to hardware not being available, but the system is ready to be connected to any direct modulator that is required to be measured upon.

The finished system is delivered as a standalone system programmed onto the Altera DE3 board, as well as source code and programming files. The source code will not be presented in the thesis due to the several thousand lines of code. Due to automatically generated code and complex simulations, the source code consists of around 6100 files, using 500 MB of disk space.

4.2 Future work

As stated earlier in the thesis, there is room for extensions of the system to provide additional tools of evaluation. The two major tracks of extended functionality are Phase correction and Predistortion. Another track is to design the opposite module, the demodulator.
4.2.1 Predistortion

In certain circumstances one might want to adjust the individual constellation points with different amplitude adjustments. One situation would be when there is an imbalance in the modulator which only affect certain constellation points and therefore are hard to adjust using the dynamic power or phase adjustments. With the help of a predistortion block, the individual constellation points can be adjusted before conversion into the analog domain.

The individual adjustment could easily be implemented using two matrices with adjustment data for the I and Q signals respectively, and adjusting these matrices using a computer since there is limited abilities of user input on the Altera DE3 board. The actual predistortion block has not been implemented but a proof of concept has been implemented to prove the possibility of such a block and the usefulness. The implementation of such a block with a user interface would be very useful.

4.2.2 Phase Adjustment

In a modulator the two paths for the I and Q signal are meant to be of the exact same length to avoid any phase shifts. Due to physical hardware limitations this is not always possible and in most modulator there is a slight delay between the two signals causing a slight phase shift. Adjustments of this property is easily achieved in the digital domain, but a complex task in the analog domain. The ability to adjust the phase is outside the scope of this thesis, but through extended work it is possible to implement such functionality.

During the conversion into an analog signal, the conversion board reads from certain registers in a sample rate specified by the Altera DE3 board. In the thesis, the sample rate has been 200 MSps. This means that a new value on the analog signal can be provided every 5 ns. By simply introducing a memory buffer in the FPGA of the Altera DE3 board, a delay of either the I or Q signal can be introduced. By introducing a single storage for the I value but not the Q value the arrival time of the signal will differ by 5 ns. This can be extended several times, depending on available hardware and resources within the FPGA.

Figure 4.1 describes how this adjustment can be seen in the analog domain, in the digital domain and a simple hardware schematic of the basic functionality. The effect of this implementation will be the delaying of the I analog signal, trying to compensate a delay of the Q signal in a modulator to get the two separate signals to merge without any phase imbalance in the modulator.

4.3 Insights

During the project, cyclic code development was stumbled upon several times. Code that was written during several days could easily be rewritten in a matter of hours in a much more effective design. The design also changed during the life cycle of the thesis, improving performance and reducing complexity by making small or major changes. During the whole project, the number of lines of final code does not come close to the actual written lines of code that have been reevaluated, improved and discarded. As the design grew bigger and more complex, the code shrank and became more sophisticated by using different techniques and re-evaluation of the design several times.
Figure 4.1: Phase adjustment theory

The system was seen as finished several times, but as measurements and verification started it became clear that there were still flaws with the system. As a finished product, the system now performs on par with expectations and it is with great pride the system is declared as fully functioning.

4.4 Final words

During the process of developing the building blocks and then combining these into the complex measurement system, my development skills have improved greatly. The knowledge I have gathered during my years at Linköping University proved really useful as the project had much in common to what I have been studying but as the system was more complex and had more strict requirements than any of my previous projects, a great deal of experience was gained. Finishing this thesis feels like a good achievement and work effort finally being acknowledged and I feel very satisfied with the work that I have done. I greatly appreciate all the help I have been given here at Chalmers Microwave Electronics Laboratory and hopefully, this work will be useful to their research as I now leave it in their hands and finish the work on this thesis.
Bibliography


Index

1024-QAM, 22, 42
128-QAM, 22, 40
16-QAM, 22, 24, 37, 47
256-QAM, 22, 41, 49
32-QAM, 22, 38
64-QAM, 22, 24, 39, 48
8-QAM, 22, 36

Altera, 11
Altera DE3, 12
Amplitude imbalance, 8
Amplitude shift keying, 5

Background, 1
Buffer, 18
Button, 26

Codeword, 5
Conversion board, 12, 19
Cross Correlation, 3
Cross correlation, 16

D/A Converter, 4
D/A converter, 12, 19
Data collection, 33
Decode, 5
Discussion, 54
Division, 26

Fourier transform, 3, 17
FPGA, 11, 12
Frequency shift keying, 5
Future work, 54

IF, 4
Issues, 51

LED, 26
Limitations, 10

Matlab, 11, 20

Measurement system, 24, 31
Modelsim, 12, 21
MODEM Project, 1
Modulation, 4, 25
Modulator, 4, 19, 34
Modulators, 30
Multiplication, 26

Oscilloscope, 14, 31
Output level, 26, 43, 51
Overshoot, 51

Phase imbalance, 8, 55
Phase shift, 20
Phase shift keying, 5
Power level, 26
PRBS detector, 16, 30, 34
PRBS generator, 15, 30, 34
Predistortion, 28, 50, 51, 55
Pseudorandom bit sequence, 15
Purpose, 2
Pushbutton, 32

QAM modulation, 7, 22
QPSK, 20, 24, 34, 46
Quartus II, 11

Results, 30, 51
RF, 4

Sampling clock, 32
SignalTap, 34
Software defined radio, 4
Switch, 26, 32

Technologies, 3

Verilog, 11, 20
Copyright

Svenska

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 enskila kopior för enskilt bruk och att använda det oförändrat för ickekommersiell 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 åktheten, säkerheten och tillgängligheten finns det lösningar av teknisk och administrativ art.

Upphovsmannens ideella rätt innefattar rätt att bli nämd 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 krävande för upphovsmannens litterära eller konstnärliga anseende eller egenart.

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

English

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 your 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/

© Josef Hederström
Linköping, November 29, 2010