The design of a high speed topology for a QPSK demodulator with emphasis on the synchronization algorithms needed for demodulation.

By

Samuel Booysen

Thesis presented in partial fulfilment of the requirements for the degree of

Master of Science in Engineering at Stellenbosch University

Supervisor: Prof. Johann Brink de Swardt

Department of Electrical and Electronic Engineering

March 2010

Declaration

By submitting this thesis electronically, I declare that the entirety of the work contained therein is my own, original work, that I am the owner of the copyright thereof (unless to the extent explicitly otherwise stated) and that I have not previously in its entirety or in part submitted it for obtaining any qualification.

March 2010

Copyright © 2010 Stellenbosch University
All rights reserved

Abstract

This thesis describes the design and implementation of a software based QPSK demodulator with a demodulation speed of 100 Mbps. The objective of the thesis was to identify a topology for the QPSK demodulator that would allow for high data rates and the design of the synchronization algorithms for carrier and symbol recovery. The QPSK demodulator was implemented on an Altera Stratix II field programmable gate array (FPGA), which does complex I and Q sampling on a down converted 720 MHz QPSK signal. The I and Q down converted baseband signals are sent through matched filters which are implemented with discrete components to maximize the signal to noise ratio of the received rectangular baseband pulses. A 1 GSPS direct digital synthesizer (DDS) is used to generate the synchronous clock for the analog to digital converters which samples the matched filter outputs. The demodulator uses two samples per symbol to demodulate the QPSK signal. A dual locking system is implemented to have a wide pre-locking filter for symbol synchronization and a narrow band post-lock filter to minimize the loop noise. A symbol lock detection algorithm decides when the symbol recovery loop is locked and switches between the loop filters.

A second 1 GSPS DDS output is mixed with a local oscillator to generate the 1.44 GHz LO signal for the quadrature down conversion. The carrier recovery loop uses a numerically controlled oscillator inside the FPGA for initial carrier acquisition which allows for very wide locking bandwidth. After lock is achieved, the external carrier recovery loop takes over and removes any frequency offset in the complex baseband signal by changing the frequency of the DDS. A QPSK modulator was also developed to provide a QPSK signal with known data. The modulator can generate any constellation diagram up to 256 points.

Opsomming

Hierdie tesis bespreek die ontwerp en implementasie van ′n sagteware gebaseerde QPSK demodulator met ′n demodulasie spoed van 100 Mbps. Die doelstelling is om ′n topologie te identifiseer vir ′n QPSK demodulator wat ′n hoë datatempo sal toelaat en ook om sinkronisasie algoritmes te ontwikkel vir draer en simbool herkenning.

Die QPSK demodulator is geïmplimenteer op ′n Stratix II FPGA van Altera wat kompleks basisband monstering doen op infase en kwadratuur basisband seine. Die basisband seine word gegenereer van ′n 720 MHz QPSK sein met ′n kwadratuur menger wiese uittrees deur puls passende filters gestuur word om die sein tot ruis verhouding te maksimeer. ′n Een gigamonster per sekonde direk digitale sintetiseerder (DDS) is gebruik om die klok vir die analoog na digitaal omsetters te genereer vir sinkrone monstering van die pulse passende filter uittrees. Die demodulator gebruik twee monsters per simbool om ′n QPSK sein te demoduleer. ′n Tweevoudige sluit algoritme word gebruik vir die simbool sinkronisasie waar ′n wyeband filter die inisiële sluit funksie verrig en dan word daar oorgeslaan na ′n nouband filter vir fase volging wat die ruis in die terugvoerlus verminder. Daar is ′n simbool sluit detektor wat identifiseer wanneer die simbool beheerlus gesluit is en selekteer dan die gepaste filter.

′n Tweede DDS en ′n sintetiseerder se uittrees word gemeng om ′n 1.44 GHz draer te genereer vir kohurente frekwensie translasie in die kwadratuur menger. Die draer sinkronisasie gebruik ′n numeries beheerbare ossilator vir die inisiële frekwensie en fase sluit wat baie vinnig geimplenteer kan word omdat dit alles in sagteware binne in die FPGA gebeur. Na die interne draer beheerlus gesluit is, neem die eksterne beheerlus oor om enige fase of frekwensie afsette in die kompleks basisband seine van die kwadratuur menger te verwyder deur die frekwensie van die draer DDS te beheer. ′n QPSK modulator is ook ontwikkel om verwysings data te genereer. Enige konstelasie vorm tot 256 punte kan geimplementeer word.

Acknowledgements

I would like to thank my wife Juanita who's love, support and understanding gave me the motivation to complete this research project.

There are numerous people I would like to thank for their contributions toward the completion of this project.

I would like to express my sincere gratitude to Prof. J.B. de Swardt for his patience, guidance and advice.

I would like to thank my colleagues at ETSE Electronics for their advice and support.

Special thanks to my whole family for their sacrifices especially during the last couple of weeks of this project for the help with our baby Ferdi.

Contents

Abbreviations

- AC - Alternating Current
- ADC - Analog to Digital Converter
- AWGN - Additive White Gaussian Noise
- BPSK - Binary Phase Shift Keying
- DAC - Digital to Analog Converter
- DC - Direct Current
- DDS - Direct Digital Synthesizer
- DSP - Digital Signal Processing
- FPGA - Field Programmable Gate Array
- IF - Intermediate Frequency
- ISI - Inter Symbol Interference
- MBPS - Mega Bits Per Second
- MSPS - Mega Sample Per Second
- NCO - Numerically Controlled Oscillator
- PED - Phase Error Detector
- TED - Timing Error Detector
- QPSK - Quadrature Phase Shift Keying

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Problem Statement

High speed digital communication has become an integral part of every day life. The demand for faster communication is insatiable and research into the field of digital communication is an ongoing process. The aim of this thesis is to design a software based QPSK demodulator with a demodulation speed of 100Mbps. The focus of this project is to identifying a topology that allows high data rates and also the development of synchronization algorithms.

1.2 Proposed Solution

The proposed QPSK demodulator will be implemented with an Altera Stratix II FPGA which will use two analog to digital converters to implement complex baseband sampling on down converted I and Q baseband signals. There is a trend in the DSP community to implement as much of the demodulation process as possible in the digital domain. This is commonly referred to as "Software Defined Radio" design. Software is easily reconfigurable which means that one could with a single hardware platform implement variable data rates and different modulation schemes without changing the hardware. Another advantage of implementing most of the demodulation in software is that once the algorithms are developed, it could be later implemented on faster platforms. As the speed of new digital hardware increase, so will the demodulation process. Thus there is a long term investment involved in developing digital algorithms instead of analog hardware.

1.3 Thesis Outline

One of the aims of this thesis was to give a logical understanding of how a QPSK modulator and demodulator functions. At the beginning of each chapter a system overview is given and highlights the hardware components that will be discussed in that particular chapter. The complete QPSK modem is shown in figure 1.1.

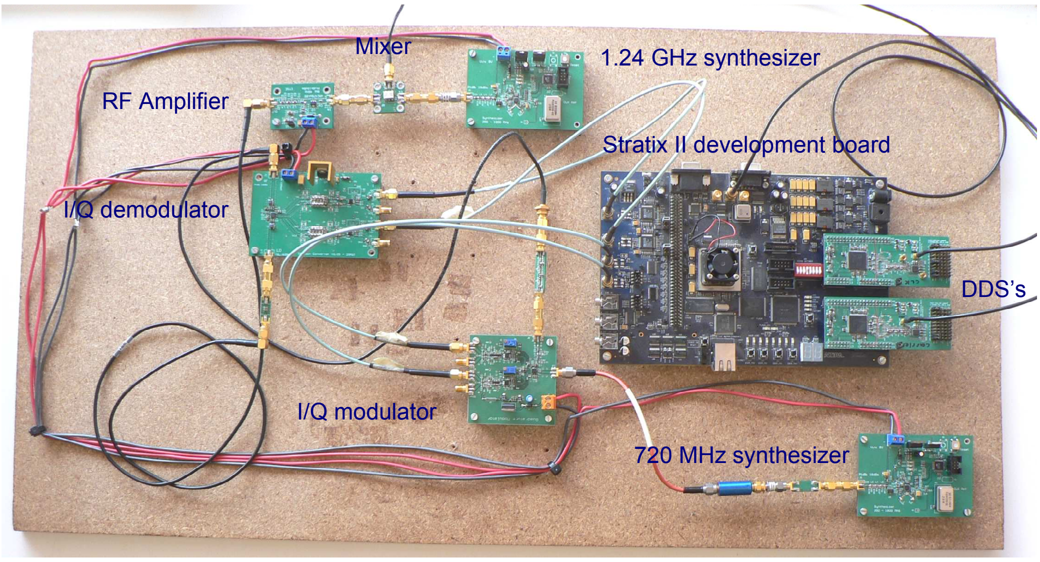


Figure 1.1: QPSK modulator and demodulator hardware.

The thesis first discuses the baseband timing recovery process followed by timing recovery simulation results. Next, QPSK signal representation and modulation is examined. A short discussion on non ideal effects in the modulation process is given. The following chapter is on the QPSK demodulation process. System simulations are done next to illustrate the operation of the demodulator. Measurements of the hardware verify the implementation. Finally, the conclusion and recommendations are given.

Chapter 2

Background

Communication systems have various forms and implementations. The common purpose of all communication systems is the transfer of information from a source through a channel to a receiver. A typical digital communication system is shown in figure 2.1. The source data is first encoded to enable error detection and possibly error correction. The encoded data is then modulated into an efficient form for transmission through a communication channel. The modulated signal is frequency translated to the transmission band before being transmitted. At the receiver it is amplified and down converted to an intermediate frequency (IF) before being demodulated. The demodulator extracts the necessary timing information needed for demodulation and then demodulates the baseband signal. The recovered binary data is decoded, checked for errors and corrected if possible.

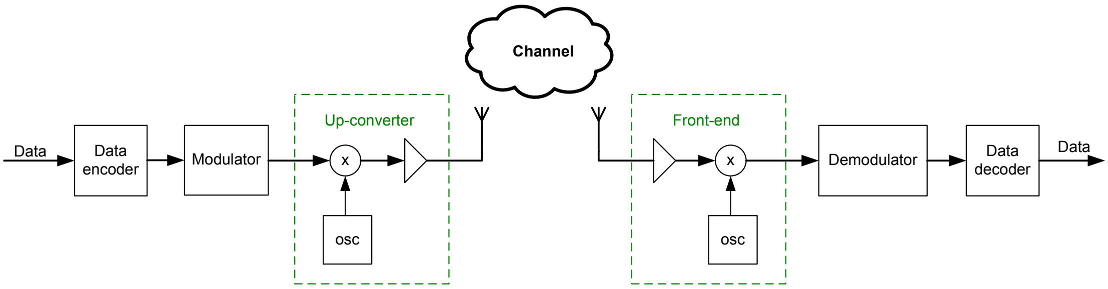


Figure 2.1: Digital communication system

The focus of this thesis lies in the design of the modulation and demodulation blocks in a communication system. Synchronization in the demodulator was the main design challenge where as the modulator was mainly designed as a data source for the development and debugging purposes of the demodulator. Synchronization in the demodulator consists of carrier and symbol timing synchronization. Baseband timing recovery will first be discussed and later shown how carrier recovery is integrated to form the complete QPSK demodulator.

Chapter 3

Baseband signal recovery

3.1 Overview of recovery process

In this section baseband bipolar signal recovery will be discussed to form a base for passband QPSK signal representation and recovery. The baseband signal is a binary pulse code modulation (PCM) signal. Refer to [1, chapter 2] for a detailed mathematical analysis on the topic of baseband synchronization. The purpose of baseband timing recovery is to determine when a transmitted symbol started and stopped. A binary PCM signal is described as shown in equation 3.1.1.

$$s(t) = \sum_i a_i g(t - iT_s) \tag{3.1.1}$$

In the above equation, a_i is real valued and represents the binary data symbols with symbol period T_s. g(t) is the pulse shape of the transmitted signal. The basic baseband receiver has the block diagram shown in figure 3.1.

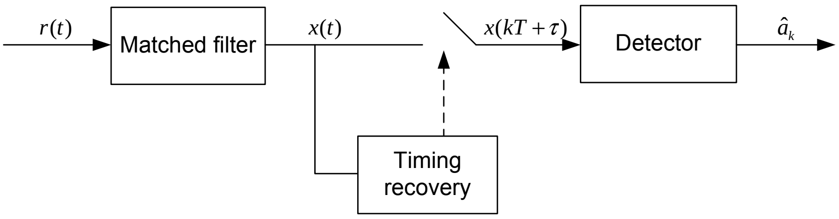


Figure 3.1: General baseband demodulation

At the receiver, the received signal with noise is shown in equation 3.1.2.

$$r(t) = \sum_i a_i g(t - iT_s - \tau) + w(t) \tag{3.1.2}$$

To demodulate this baseband signal, the unknown parameters T_s and τ need to be estimated. The time delay τ is estimated with a timing recovery loop. The transmitter symbol period T_s is usually well specified, but there will always be a slight offset in frequency when compared to another clock source at a receiver. It will however be assumed that any small frequency offset can be compensated for by the timing recovery loop. The noise at the receiver is assumed as additive white Gaussian noise (AWGN) with double sided noise spectral density of $\frac{N_0}{2}$ W/Hz.

The first step in designing the demodulator is to decide what type of demodulator topology to use. The demodulation process can be done synchronously or asynchronously. What is implied by synchronous demodulation is that the analog to digital converter's clock is controlled in frequency and phase to ensure that exactly N samples are sampled per bit. Typically systems try to sample in the middle (peak) of the transmitted pulses and at the transitions. The analog to digital converter's clock frequency must be an even integer multiple of the received signal's symbol rate and the phase of the ADC clock transition must coincide with the middle of the received pulses. This implementation is a hybrid continuous-time/discrete-time approach to symbol timing recovery as shown in figure 3.2.

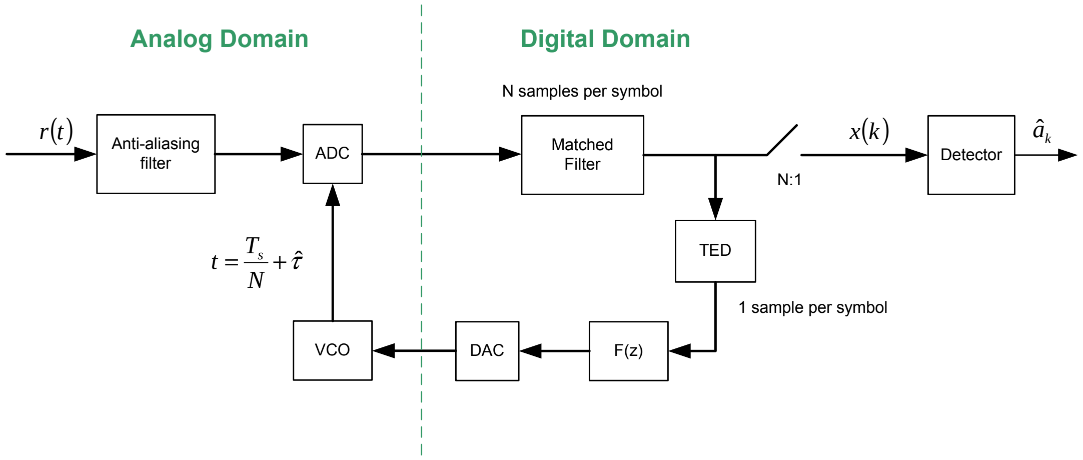


Figure 3.2: Synchronous baseband demodulation

The asynchronous recovery method is a discrete-time method which means that the received signal is sampled with a fixed clock and an interpolation algorithm is used to estimate samples between the sampled values which coincide with the peak of the symbol pulses. See figure 3.3

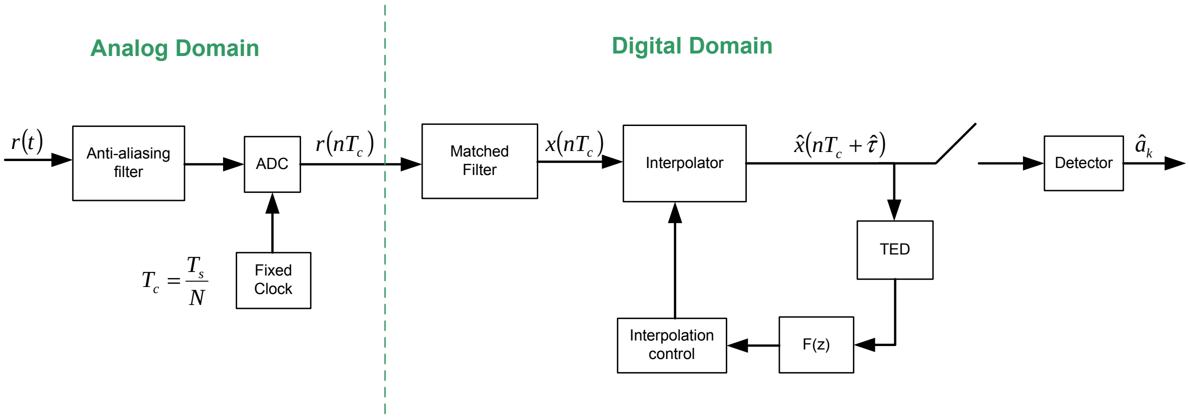


Figure 3.3: Asynchronous baseband demodulation

There are many advantages and disadvantages for both methods that must be considered. They are summarized as follows:

Synchronous demodulation advantages:

1. Digital component of demodulator is easier to implement than asynchronous demodulation.

Synchronous demodulation disadvantages:

1. Complicated analog hardware is needed to transfer the timing information from digital domain to analog domain.

2. The matched filter in the feedback path of the recovery loop introduces a transport delay which increases the response time of the recovery loop.

3. This system will also have higher phase noise or timing jitter since the clock of the ADC needs to be variable which normally suffer from higher phase noise than fixed clock oscillators.

Asynchronous demodulation advantages:

1. This process is an all digital implementation which means that no additional analog hardware is needed. This also makes simulation of the recovery process easy to implement.

Asynchronous demodulation disadvantages:

1. The interpolator in the recovery loop is quite complex and processing intensive. An interpolator usually requires a large number of sequential operations that could severely limit the maximum demodulator speed as an interpolant is needed for each ADC sample. There are pipelined interpolators such as Farrow interpolators which has a pipelined structure that will allow for higher clock rates at the cost of extra transport delay.

2. Another disadvantage is the interpolation jitter which occurs when the ADC sampling rate is not a even integer multiple of the symbol rate. This problem makes synchronous retransmission of the data difficult to implement.

Taking the above into consideration, it was chosen to implement synchronous demodulation. The biggest drawback of this method is the high phase noise of the voltage controlled oscillator (VCO). To solve this problem, a direct digital synthesizer (DDS) was selected to replace the VCO as seen in figure 3.4. The DDS has a much lower phase noise than a regular LC tuned VCO. The reason for this is because the output of the DDS is derived from a crystal reference clock.

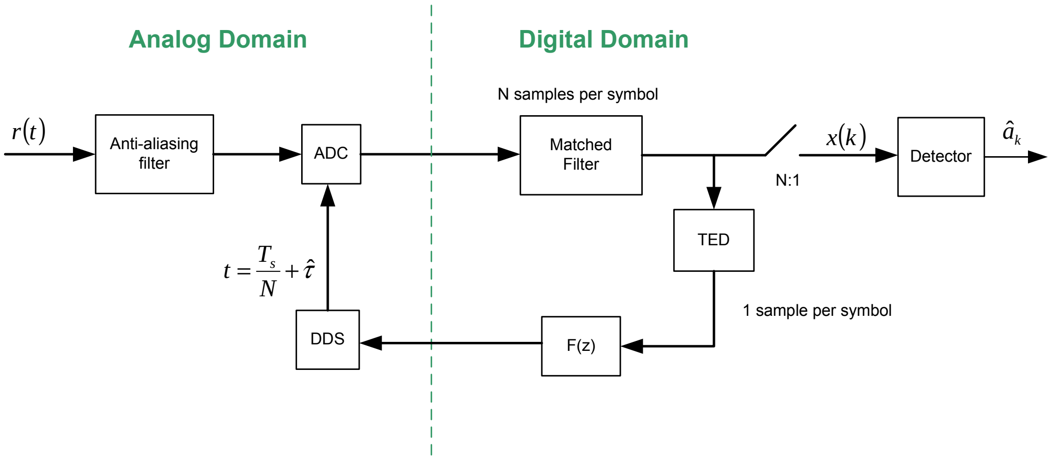


Figure 3.4: Synchronous baseband demodulation with DDS clock

Selection of the ADC clock frequency and thus the number of samples per symbol available is a very important decision which has far reaching consequences on the demodulation algorithm of the QPSK demodulator. The maximum sampling rate of the ADC's on the FPGA development board is 120MHz and the demodulator data rate specification is 100Mbps. In a QPSK system, the symbol rate is half of the bit rate. Due to the restriction of the maximum sampling rate of the ADC's, a two sample per symbol demodulator is implemented. The ADC's is thus clocked at 100MHz. With two samples per symbol, a matched filter can not be implemented in the digital domain. The only alternative is to implement the matched filter in the analog domain. See figure 3.5.

The next critical block to design is the timing error detector (TED). The function of the timing error detector is to determine where the middle of each symbol is. The TED block generates an error signal to increase or decrease the frequency of the DDS generated clock. The output of the timing error detector goes through a loop filter which sets the damping factor and loop bandwidth. The design of the loop filter requires the development of a transfer function for the recovery loop which means every block in the recovery must be mathematically described.

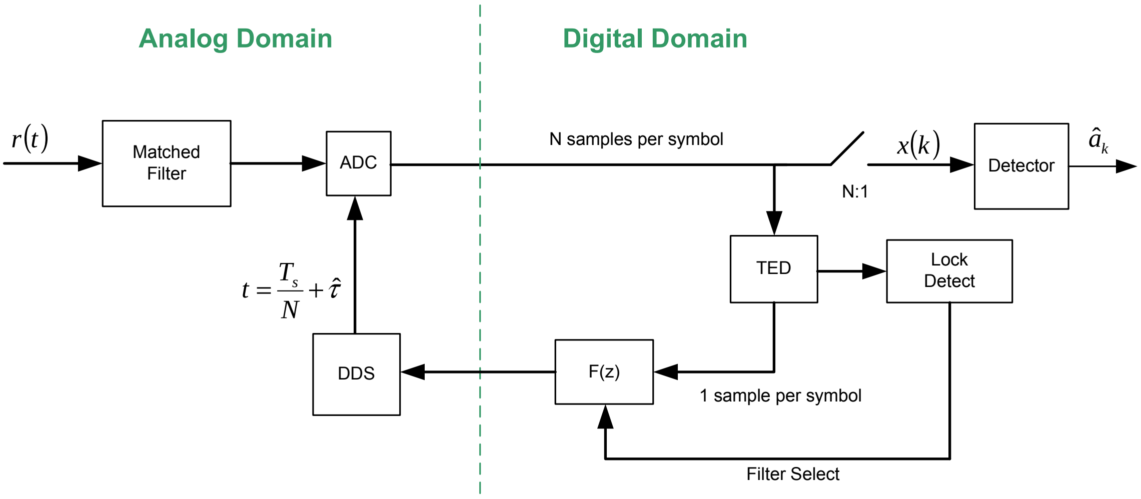


Figure 3.5: Synchronous baseband demodulation with matched filter

There is a timing lock detector that determines when the recovery loop is locked and switches from a pre-lock filter to a post-lock filter. The detector block is a level comparator to identify the transmitted symbol.

Now that an overview of the recovery process has been given, the design of each block in figure 3.5 is discussed in the rest of this chapter.

3.2 Matched filter

The purpose of the matched filter is to maximize the signal to noise ratio of a received pulse and is the first block in the baseband receiver. See figure 3.6. The matched filter is the optimum receive filter to implement for the detection of a pulse in additive white Gaussian noise (AWGN). This is done by cross-correlating the received signal with a known noiseless version of the received pulse [2]. It was assumed that rectangular pulses will be received by the demodulator. The impulse response of the matched filter is thus also a rectangular pulse. The receive filter is related to the transmit filter as shown in equation 3.2.1.

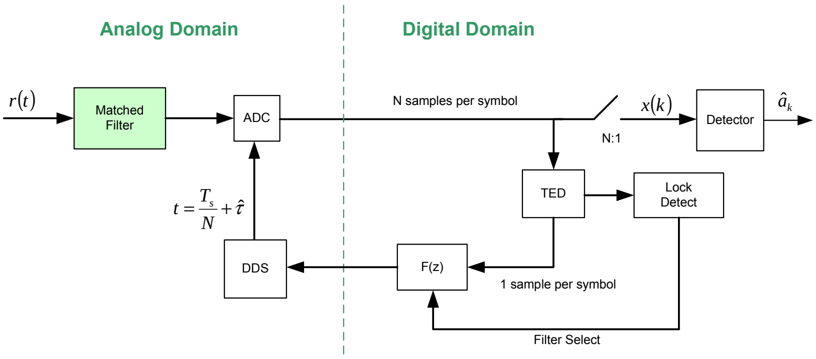


Figure 3.6: Synchronous baseband demodulation with DDS clock

$$g_m(t) = kg^*(t_m - t) \tag{3.2.1}$$

In equation 3.2.1, the constant k just scales the filter output and will be assumed to be one. The auto correlation of a transmitted pulse is defined in equation 3.2.2. $-L_pT_s \leq t \leq L_pT_s$ is the interval over which the transmitted pulse is defined.

$$R_p(u) = \int_{-L_pT_s}^{-L_pT_s} g(t)g(t-u)dt \tag{3.2.2}$$

Given the received signal has the form shown in equation 3.1.2, the matched filter output is expressed in equation 3.2.3. $v(t)$ is the result of the cross correlation of the matched filter impulse response with the received signal's noise.

$$x(t) = \sum_i a_i R_p(t - iT_s - \tau) + v(t) \tag{3.2.3}$$

Figure 3.7 shows the matched filter input and output. When the noise power spectral density is assumed $S_n(\omega) = N_o/2$, the mean square noise at the output of the filter is expressed in equation 3.2.4 [3]. The noise at the output of the filter is a function of the noise spectral density and the

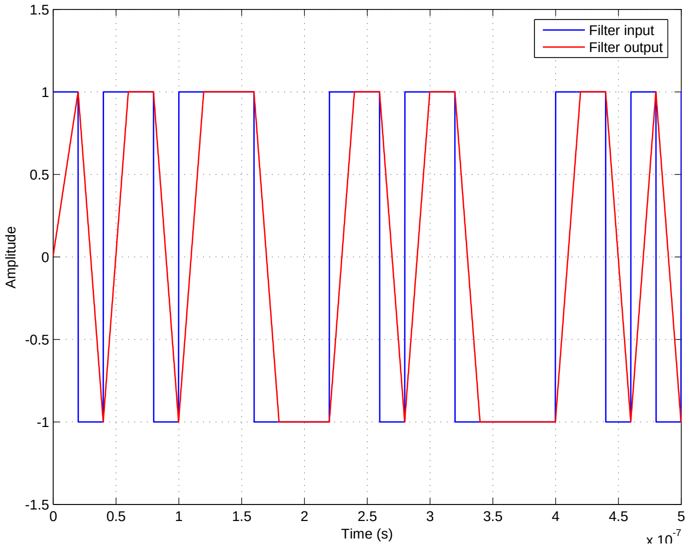


Figure 3.7: Matched filter input and output.

energy of the filter.

$$\sigma_n^2 = \overline{v^2(t)} = \frac{N_o R_p(0)}{2} \tag{3.2.4}$$

The matched filter output reaches a maximum equal to the pulse energy at an abscissa $t = t_m$. The signal energy to noise power spectral density is shown in equation 3.2.5 where a_i refers to the symbol amplitude.

$$\eta = \frac{a_i^2 T_s}{N_o} = \frac{E_s}{N_o} \tag{3.2.5}$$

The matched filter was design with the aid of a CAD program named Filter Solutions 2006. The bit rate of the baseband data is 50 Mbps which has a period of 20 ns. Ideally one would want to design a filter whose impulse response matched to the 20 ns pulse width of the transmitted pulses. This implies that the step response of the filter would also have a 20 ns rise time. Due to rounding of components to industry available values, component tolerance and slight rounding of the transmitter pulse shape at the transitions, the matched filter output has slight intersymbol interference. When there is no transition from the previous bit, the symbol amplitude is slightly higher than for the case where it was preceded by a transition. Reducing the rise time would solve the problem but it also increases the out of band noise. A compromise was made and a balanced 9th order filter was designed with a rise time of 18 ns. The order of the filter determines how linear the slope of the matched filter output will be as well as how sharp the tip at the maximum of the output is. Measurements will show that the resulting rise time of the matched filter pulses are very close to 20 ns. Figure 3.8 shows the implemented filter schematic diagram with standard component values.

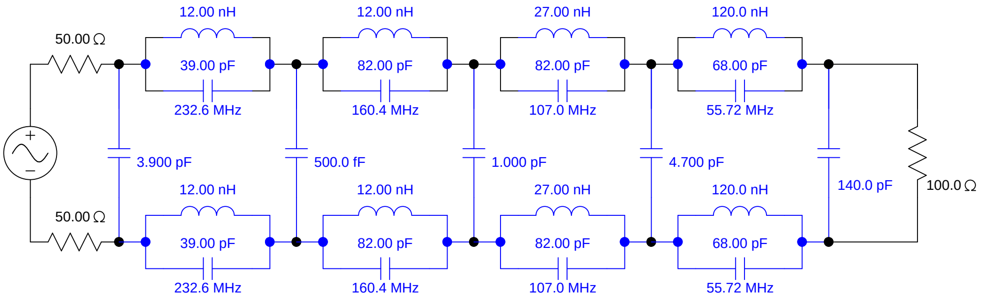


Figure 3.8: Matched filter schematic.

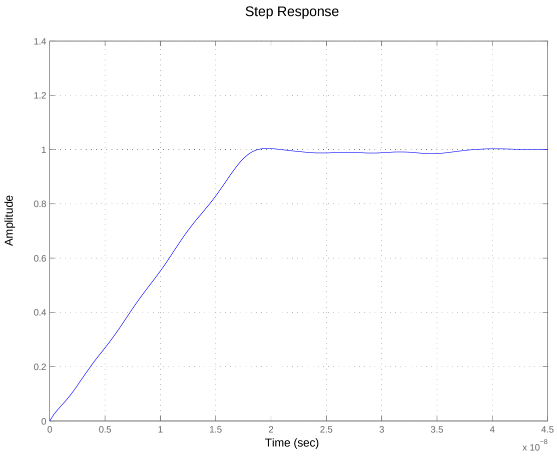


Figure 3.9: Matched filter step response.

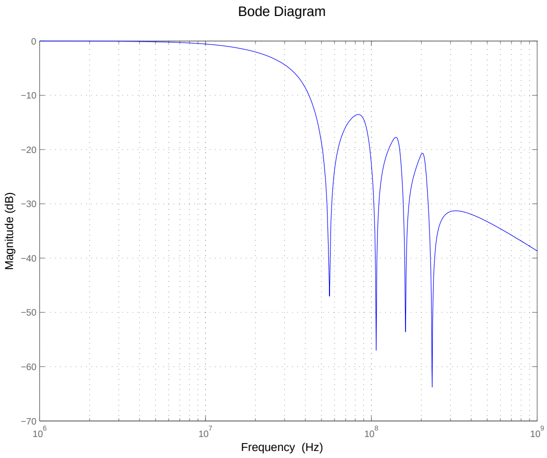


Figure 3.10: Matched filter magnitude response.

3.3 Analog to digital converter

The analog to digital converter transfers the matched filter's output from the analog domain to the digital domain. The analog to digital converters used in the demodulator was the FPGA development board's on-board ADC's. The ADC's are the AD9433 from Analog Device which are 12 bit ADC's with a maximum conversion rate of 125MSPS. The quantization error for an analog to digital converter with symmetrical uniform intervals produces a mean square distortion as shown in equation 3.3.1 [4]. $f_X(x_i)$ represents the probability density function of the analog signal and Δ the quantization interval.

$$D = \frac{\Delta^2}{12}\sum_{i=1}^{N} f_X(x_i)\Delta \cong \frac{\Delta^2}{12} \tag{3.3.1}$$

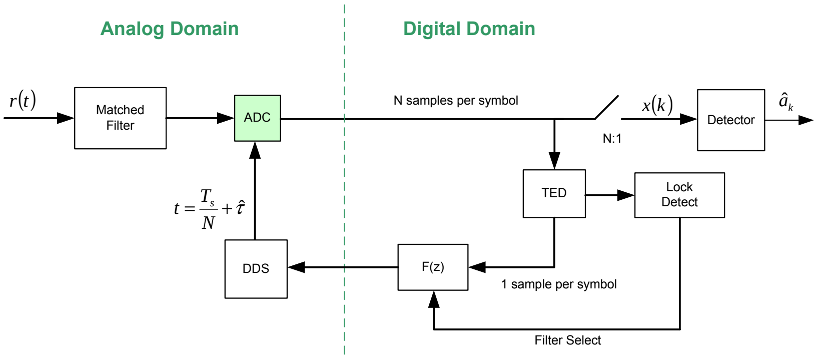


Figure 3.11: Synchronous baseband demodulation with DDS clock

A more convenient measure of the performance of an ADC is the signal to noise power ratio. A typical ADC with a range of four times the standard deviation of the input analog signal produces the equation 3.3.2.

$$SNR = 6.02n - 7.3\ dB \tag{3.3.2}$$

Thus for a 12 bit ADC and signal levels as described above, the SNR of the ADC is 64.96dB.

3.4 Direct digital synthesizer

As shown in figure 3.12, the purpose of the direct digital synthesizer (DDS) is to generate the variable clock for the ADC's. The DDS clock is the main clock that also drives almost all the demodulator logic in the FPGA. A DDS from Analog Devices was selected which has a DAC with a maximum sample rate of 1 GSPS. At the maximum clock rate, this DDS is capable of generating a clock of up to 400 MHz while leaving enough bandwidth to implement an output filter.

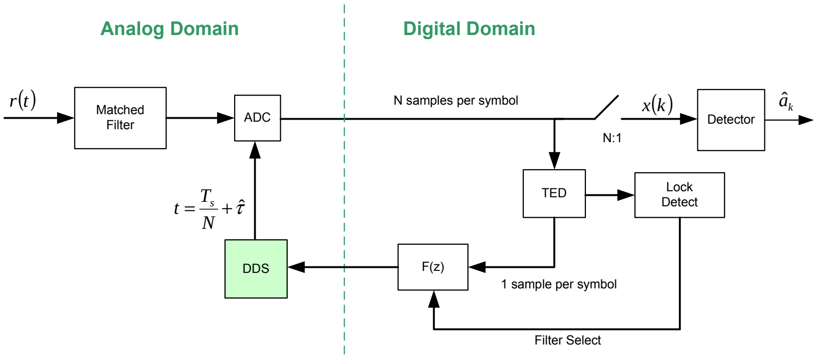


Figure 3.12: Synchronous baseband demodulation with DDS clock

The reason why one can not generate carriers up to half the DAC sample rate is due to the mixing products that are generated. Figure 3.14 shows the unwanted mixing products that are created

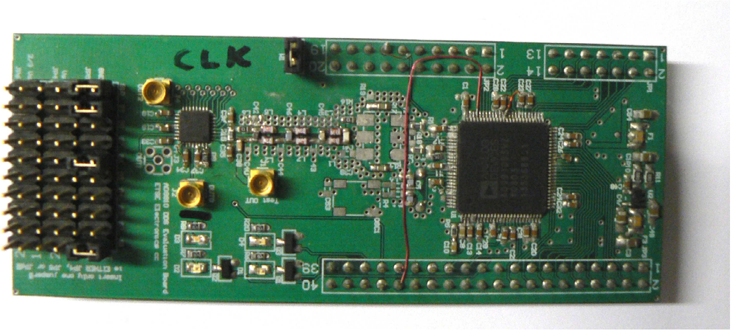


Figure 3.13: Symbol recovery DDS board.

at the sum and difference frequencies of the fundamental output frequency and the DAC clock frequency components.

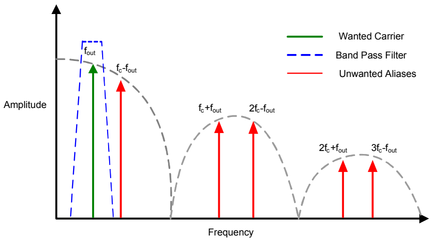


Figure 3.14: Theoretical DDS spurious content

Refer to figure 3.15 for the DDS block diagram. The maximum internal clock rate was implemented on the DDS. The DDS has a software selectable reference multiplier. A 25 MHz crystal oscillator as reference, together with the multiplier set to 40 times, generates an internal clock at 1 GHz. To generate the ADC clock, the DDS output waveform must be converted to a square wave. Before that could be done, it is first filtered to remove the aliases. The filtered sinusoid is then sent to a clock distribution IC which converts the filtered sinusoid to 3.3 V CMOS logic levels.

A NIOS II soft core processor is implemented in the FPGA to perform initial configuration and monitoring of peripherals in the demodulator. The NIOS processor uses an SPI bus to setup the DDS. After the DDS is configured, the parallel programming bus is enabled which continuously reprograms the frequency corrections into the DDS. The filters implemented were designed to allow for a clock range of 33-400 MHz. It consists of a second order high pass filter, followed by a 6th order Chebyshev low pass filter. The frequency responses are shown in figure 3.16. As mentioned in section 3.1, the DDS is programmed at a rate of 50 MHz. To reduce ringing and reflections on the programming bus, the length of the programming bus is kept to a minimum by socketing the DDS board directly onto one of the expansion ports of the FPGA evaluation board. See figure 3.17 for the DDS board schematic diagram. The frequency word that sets the output

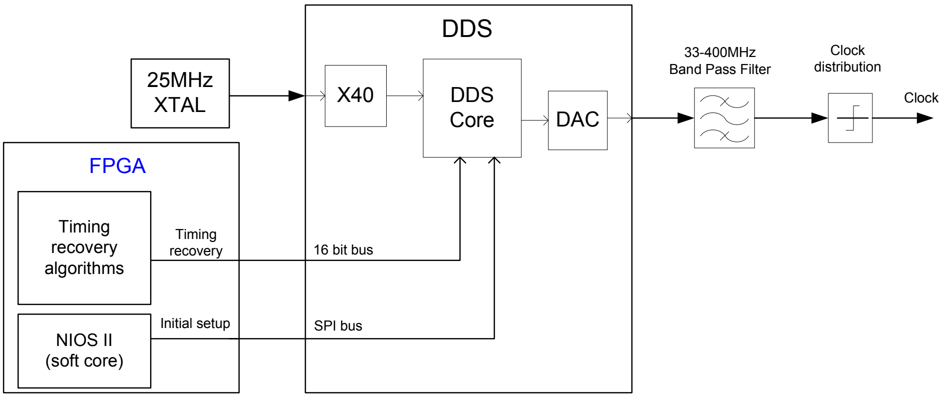


Figure 3.15: DDS block diagram

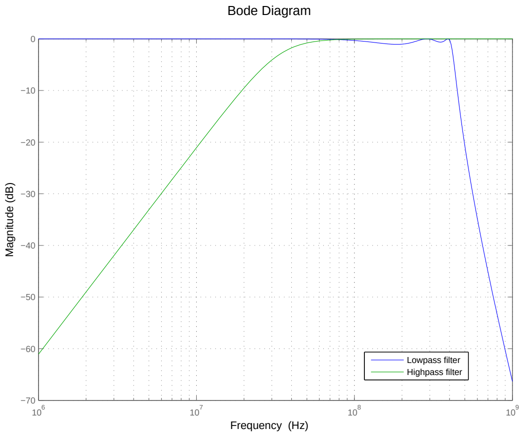


Figure 3.16: DDS filter frequency response

frequency is a 32 bit word. The parallel programming bus however is only 16 bits wide. Only 16 of the 32 bits can be programmed per cycle when the DDS is in parallel programming mode. An offset word can be programmed with the SPI bus to specify which 16 bits in the 32 bit word will be programmed in the parallel mode. This effectively sets the minimum frequency step and maximum frequency range for the parallel programming interface.

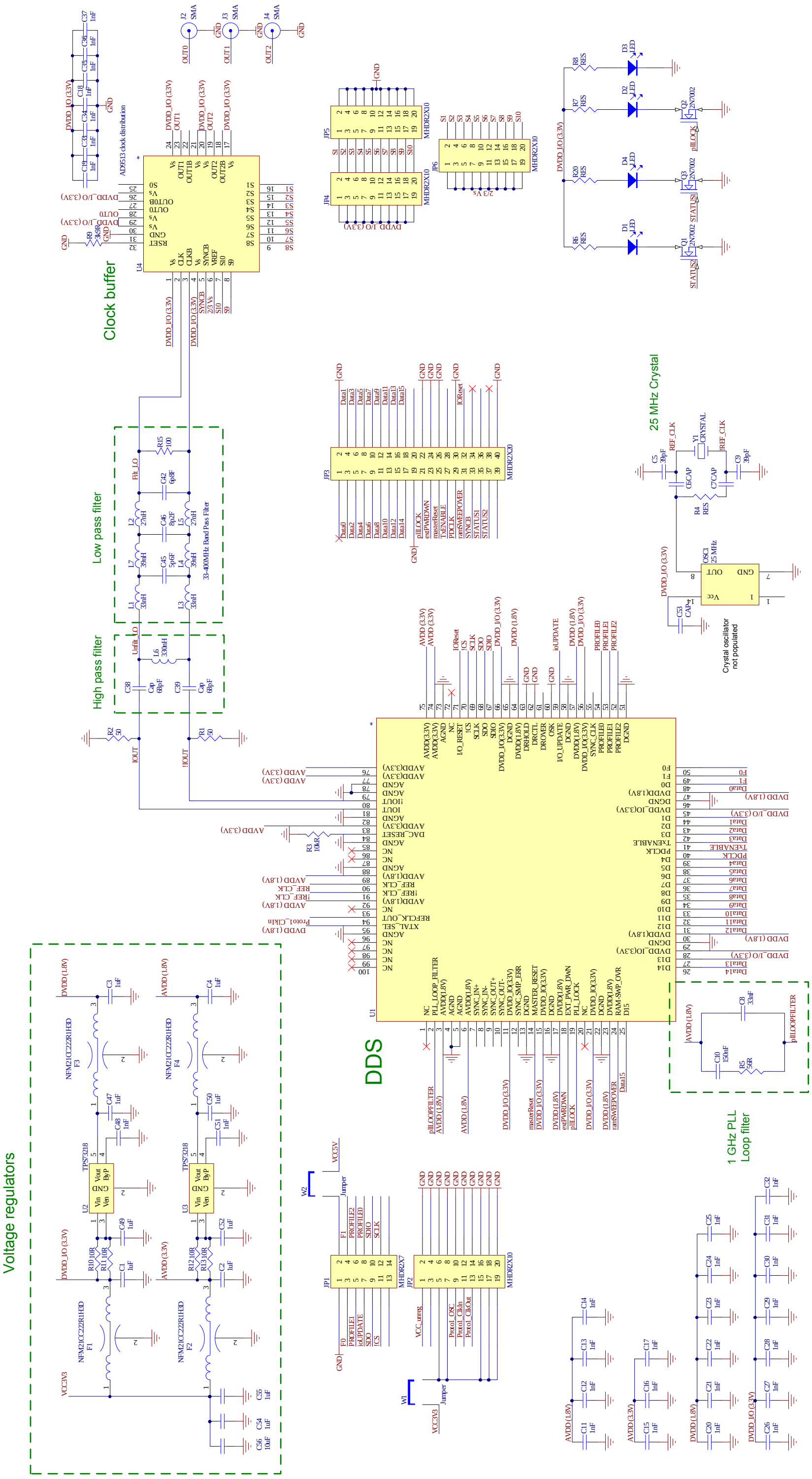


Figure 3.17: DDS board schematic

3.5 Timing Error Detector

This section explains how timing information is extracted from the sampled pulses. When a matched filter is implemented, the timing error detector (TED) is used to decide when to sample the matched filter output. Figure 3.18 shows the TED position in the timing recovery loop.

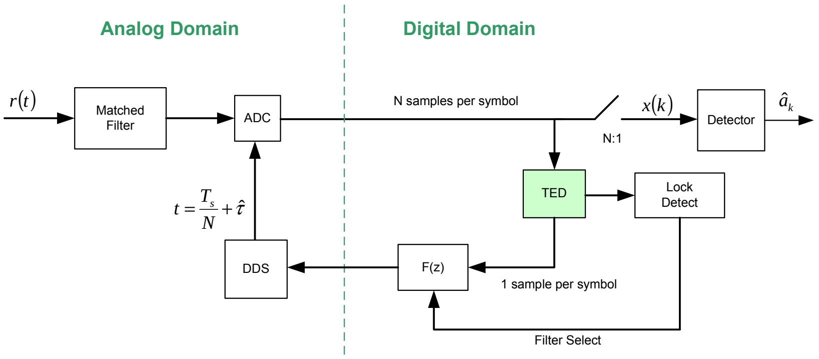


Figure 3.18: Synchronous baseband demodulation with DDS clock

There are various implementations for timing error detectors. A maximum likelihood estimator is designed in this section and is compared to other timing error detector at the end of the chapter. The maximum likelihood TED is best explained by looking at the eye diagram and derivative eye diagram of the baseband signal shown in figure 3.19.

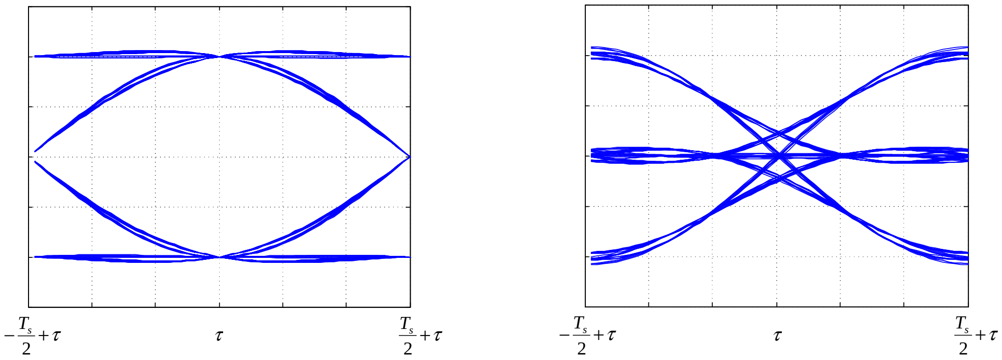


Figure 3.19: Eye diagram and derivative eye diagram

The aim of the TED is to generate an error signal to guide the DDS to sample at the maximum of the eye opening. At maximum eye opening, the slope is zero. This is also shown in the derivative eye diagram. Thus, an error signal could be generated from the derivative of the baseband signal. Also note that when there is no transition, the slope is also zero. This is to be expected since there is no transition information available. There needs to be transitions in order for the system to synchronize. The sign of the baseband signal changes, depending on the data symbol sent. This introduces a sign ambiguity when using the slope as an error signal. Consider the four cases shown in figure 3.20.

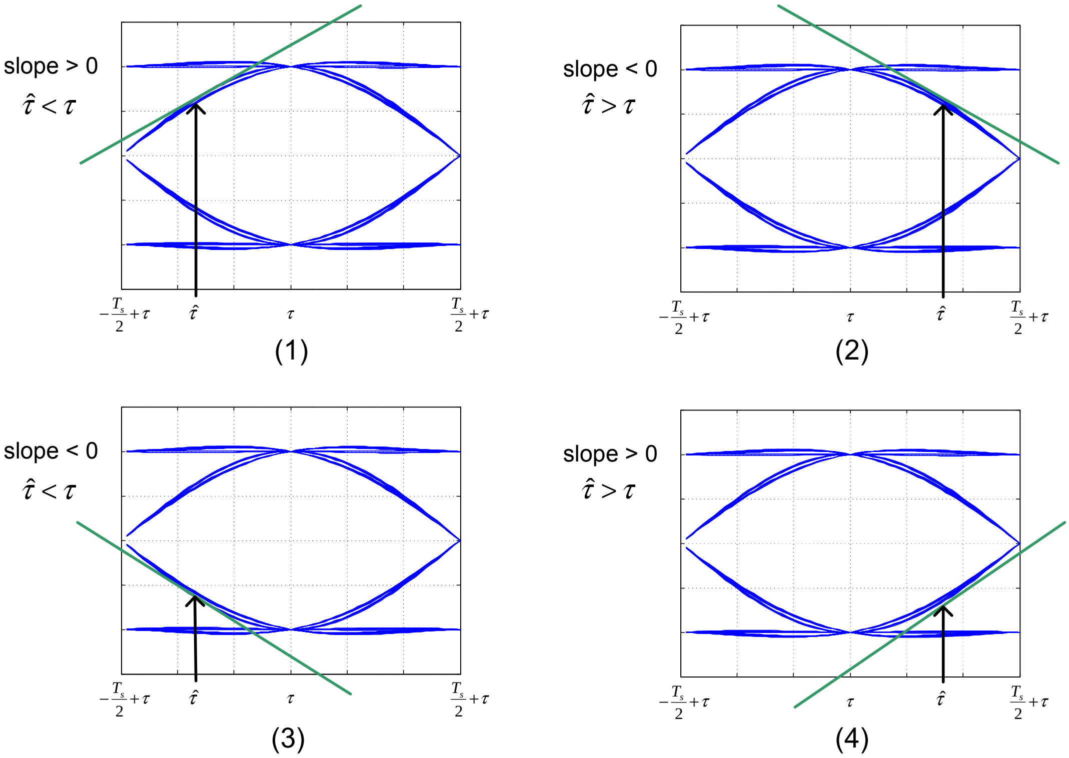


Figure 3.20: Eye diagram timing ambiguity

1. The slope is positive when the estimated timing is a bit early and the transmitted symbol is positive.

2. The slope is negative when the estimated timing is a bit late and the transmitted symbol is positive.

3. The slope is negative when the estimated timing is a bit early and the transmitted symbol is negative.

4. The slope is positive when the estimated timing is a bit late and the transmitted symbol is negative.

What is needed is a positive error signal when the estimated timing is early and a negative error signal for a late estimate. As shown in equation 3.5.1, multiplying the slope of the pulse shape with the transmitted symbol amplitude corrects the sign and is referred to as a data aided system.

$$e(k) = a(k)\dot{x}(kT_s + \hat{\tau}) \qquad (3.5.1)$$

The actual symbol is usually not available since it is the purpose of the demodulator to determine the sign. Instead, the estimate of the symbol is used. For a binary signal, the sign should be multiplied with the symbol derivative.

$$e(k) = \hat{a}(k)\dot{x}(kT_s + \hat{\tau}) \tag{3.5.2}$$

The derivative block is typically implemented as a filter. With two samples per symbol, the filters could not be implemented. A derivative can also be approximated with a difference.

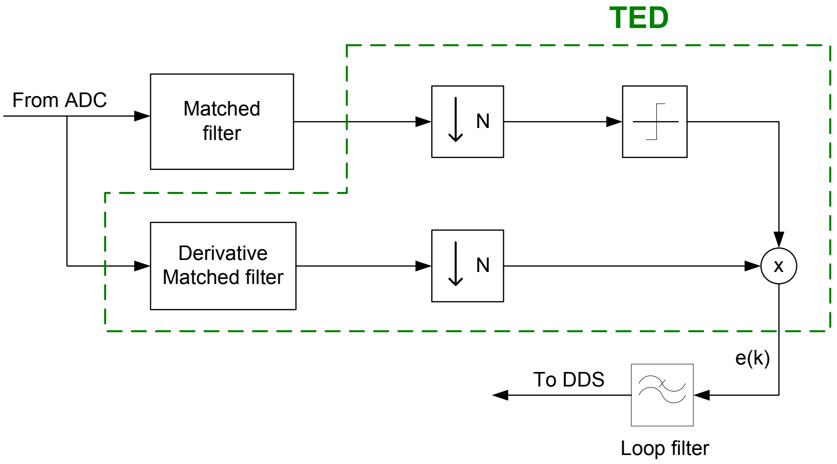


Figure 3.21: Maximum likelyhood timing error detector diagram

$$\dot{x}(\tau) = \lim_{\Delta \to 0} \frac{x(\tau + \Delta) - x(\tau - \Delta)}{2\Delta} \tag{3.5.3}$$

If this approximation is substituted into equation 3.5.2, the result is the early-late timing error detector. Setting delta to half a symbol produces the early late timing error detector for a two samples per symbol system. The block diagram is shown in figure 3.22.

$$e(k) = \hat{a}(k) \cdot [x((k + 1/2)T_s + \hat{\tau}) - x((k - 1/2)T_s + \hat{\tau})] \tag{3.5.4}$$

An important parameter of a TED is its gain. The gain is needed when designing the loop filter. The matched filter output has the following form when it is sampled by an ADC.

$$x(nT_c) = \sum_i a_i R_p(nT_c - iT_s - \tau) + v(nT_c) \tag{3.5.5}$$

The transmitted symbols are assumed to be statistically independent so that the following property is true.

$$E\{a(k)a(m)\} = \sigma_a^2(m - k) \tag{3.5.6}$$

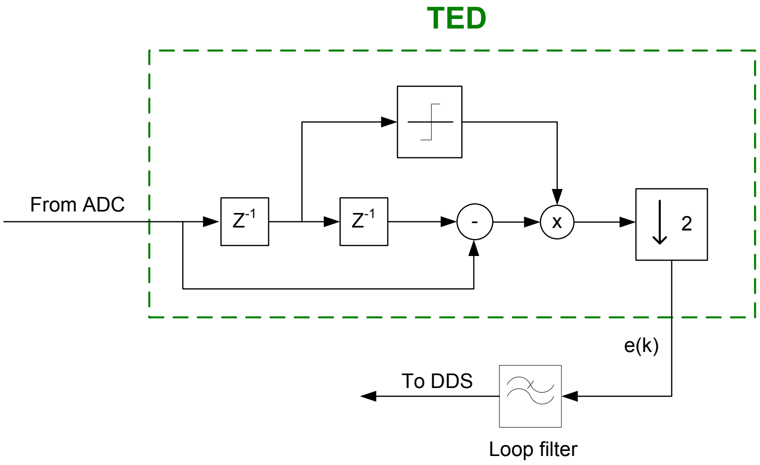


Figure 3.22: Early-late timing error detector diagram

where

$$\sigma_a^2 = E\{a^2(k)\} \tag{3.5.7}$$

To determine the gain of the TED, the S-curve for the timing error detector needs to be derived. Assume for a moment that $T_c \approx T_s$. Equation 3.5.8 shows the matched filter output expressed at the symbol rate since the TED generates the error signal at the symbol rate.

$$x(nT_s + \tau) = \sum_i a_i R_p((n-i)T_s + \hat{\tau} - \tau) + v(nT_s) \tag{3.5.8}$$

To determine the S-curve of the early late TED, the expected value is calculated. Substituting equation 3.5.8 into equation 3.5.4 and taking the expected value, produces $h_e(\tau_e)$. Assuming $\hat{a}(k) = a(k)$, $h_e(\tau_e)$ is shown in equation 3.5.9.

$$\begin{aligned} h_e(\tau_e) &= E\{\hat{a}(k) \cdot [x((k+1/2)T_s + \hat{\tau}) - x((k-1/2)T_s + \hat{\tau})]\} && (3.5.9) \\ &= E\{\hat{a}(k) \cdot [\sum_i a_i R_p((n-i+1/2)T_s - \tau_e) \\ &\quad - \sum_i a_i R_p((n-i-1/2)T_s - \tau_e)]\} && (3.5.10) \\ &= \sigma_a^2 [R_p(T_s/2 - \tau_e) - R_p(-T_s/2 - \tau_e)] && (3.5.11) \end{aligned}$$

The gain of the TED is the slope of $h_e(\tau_e)$ when $\tau_e = 0$. Evaluating equation 3.5.11, one sees that the gain is a function of the symbol amplitude and pulse shape. The derivative of the matched filter output is shown in figure 3.23. The TED gain is the difference in slope at the beginning and end of the baseband pulse. From figure 3.23, the derivative signal amplitude at the data transitions is 2.3. The difference in slope for two alternating pulses would then be 4.6. Thus the TED gain is 4.6. If there is not two consecutive transitions, the TED gain is 2.3. It should become evident that the TED gain is dependent on the data transitions and is further discussed in section 4.1.

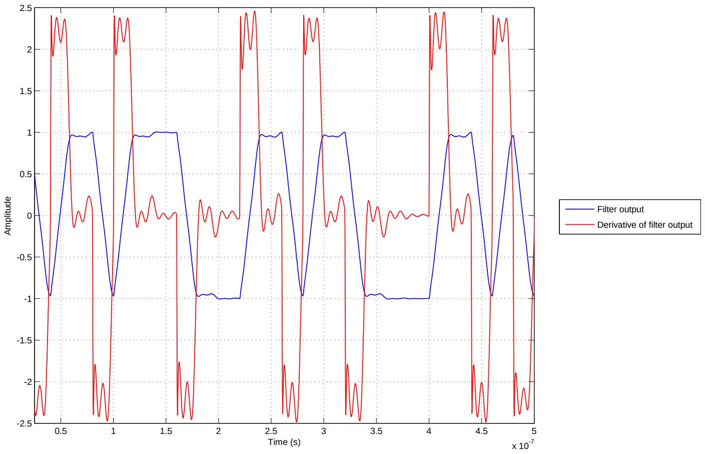


Figure 3.23: Derivative of baseband data.

A timing error detector can have only one stable lock point per bit. Stable lock points on an S-curve are zero and have a positive slope. In figure 3.24, one can clearly see that the TED output is zero at the transitions of the baseband data signal and that the slope is positive. Thus, the zero crossings in figure 3.24 are the positions that produce stable lock points in the TED. Also as mentioned earlier, the TED output is zero when there are no data transitions since no timing information is available without transitions. In the simulation model of the baseband recovery loop shown in section 4.1, there is a minus sign in the feedback loop, since negative feedback is implemented. In the practical implementation on the FPGA, the minus sign is absorbed in the TED to produce the error signal shown in equation 3.5.12.

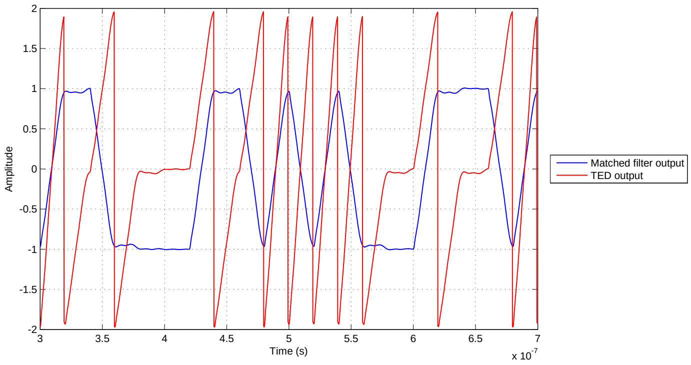


Figure 3.24: Early late timing error detector output waveform.

$$e(k) = \hat{a}(k) \cdot \left[x\left((k-1/2)T_s + \hat{\tau}\right) - x\left((k+1/2)T_s + \hat{\tau}\right) \right] \tag{3.5.12}$$

The other timing error detectors that are commonly used is the zero crossing detector and the Gardner timing error detector [5] described in equations 3.5.13 and 3.5.14 respectively.

$$e(k) = x(kT_s + \hat{\tau}) \cdot \left[\left| x\left((k-1/2)T_s + \hat{\tau}\right) \right| - \left| x\left((k+1/2)T_s + \hat{\tau}\right) \right| \right] \tag{3.5.13}$$

$$e(k) = x(kT_s + \hat{\tau}) \cdot \left[x\left((k-1/2)T_s + \hat{\tau}\right) - x\left((k+1/2)T_s + \hat{\tau}\right) \right] \tag{3.5.14}$$

3.6 Timing loop filter

All the necessary blocks in the timing recovery feedback loop have been described to determine the loop characteristics, except for the loop filter. See figure 3.25.

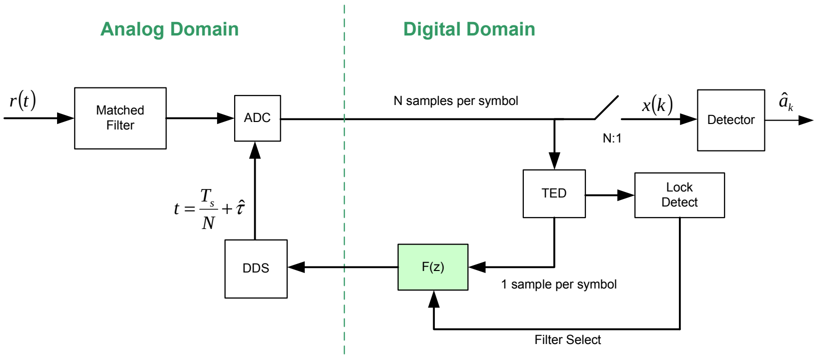


Figure 3.25: Synchronous baseband demodulation with DDS clock

The timing recovery loop can be viewed as a digital phase lock loop. A discrete time PLL has a phase detector, a discrete time loop filter and a numerically controlled oscillator. The implementation of the timing recovery loop however, has a TED as the phase detector, a discrete time filter and a DDS as the numerically controlled oscillator. See figures 3.26 and 3.27. Background information on continues time phase locks are given in appendix A and is used later to determine the loop filter coefficients. As a general guide, the continuous time PLL theory was adapted for the digital domain. As in the usual design process of PLL's, the timing recovery loop will be approximated with a linear equivalent model.

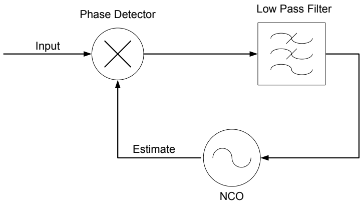


Figure 3.26: Digital phase lock loop topology

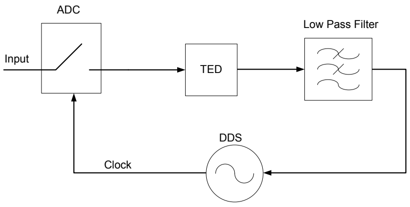


Figure 3.27: Timing recovery loop diagram

The TED gain, as derived in equation 3.5.11, is a function of the received signal amplitude and the pulse shape. As the loop gain is varied, the closed loop poles of the recovery loop move on a root locus. Ideally, the poles should be fixed, which means the loop gain and thus the received signal amplitude must be fixed. To accomplish this, some form of automatic gain control (AGC) is needed. An AGC will be introduced later in the passband system, but for the current analysis, assume the input signal amplitude is normalized to one. The gain of the TED is simply a constant (K_p).

The loop filter is determined by taking the z-transform of the continuous time loop filter shown in equation A.1.3. The resulting proportional plus integral loop filter has the z-domain representation shown in equation 3.6.1.

$$F(z) = K_1 + \frac{K_2}{1 - z^{-1}} \tag{3.6.1}$$

The DDS has a phase equivalent model shown in equation 3.6.3.

$$\begin{aligned} \hat{\theta}_{NCO}(nT) &= K_0 \sum_{k=-\infty}^{n} v(kT) && (3.6.2) \\ \hat{\theta}_{NCO}(z) &= \frac{K_0}{1 - z^{-1}} && (3.6.3) \end{aligned}$$

The phase equivalent discrete time PLL is shown below and will be used to design the loop filter constants. A unit delay in the feedback path is also necessary because the one input of the phase detector is a function of its output. The solution is to insert a delay to give time to calculate the phase detector inputs from the previous cycle before the output of the phase detector of the current cycle is calculated. Also note that the delays from the ADC and DDS have been intentionally left out. The additional delays are discussed in section 4.1 but for now the loop bandwidth is assumed to be chosen sufficiently low so that the mentioned delays has a negligible effect on the loop characteristics which will also reduce the noise in the control loop. The next step is to derive the transfer function of the discrete time PLL.

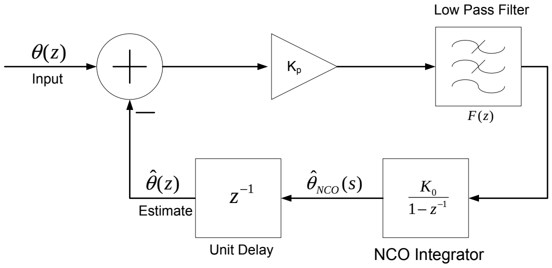


Figure 3.28: Linearised digital phase lock loop

$$H_d(z) = \frac{\hat{\theta}}{\theta} = \frac{-K_0K_pK_1z^{-2} + K_0K_p(K_1+K_2)z^{-1}}{(1-K_0K_pK_1)z^{-2} + (K_0K_p(K_1+K_2)-2)z^{-1} + 1} \tag{3.6.4}$$

The next step is to determine the constants that would yield the desired loop characteristics. Tustin's bi-linear transform is used to convert the transfer function of a continuous time PLL to its equivalent discrete time representation and compared it to the transfer function of the discrete time PLL in equation 3.6.4. Tustin's equation is shown in equation 3.6.5 and is a linear approximation to integration between two points.

$$\frac{1}{s} = \frac{T}{2}\frac{1+z^{-1}}{1-z^{-1}} \tag{3.6.5}$$

Substituting equation 3.6.5 into equation A.1.6, which is the transfer function of the continuous time PLL, yields the following result:

$$\begin{aligned}
H_a\left(\frac{2}{T}\frac{1-z^{-1}}{1+z^{-1}}\right) &= \frac{2\zeta\omega_n \cdot \left(\frac{2}{T}\frac{1-z^{-1}}{1+z^{-1}}\right) + \omega_n^2}{\left(\frac{2}{T}\frac{1-z^{-1}}{1+z^{-1}}\right)^2 + 2\zeta\omega_n \cdot \left(\frac{2}{T}\frac{1-z^{-1}}{1+z^{-1}}\right) + \omega_n^2} \\
&= \frac{z^{-2}(\omega_n^2 - \zeta\omega_n\frac{4}{T}) + z^{-1}2\omega_n^2 + \zeta\omega_n\frac{4}{T} + \omega_n^2}{z^{-2}(\frac{4}{T^2} - \zeta\omega_n\frac{4}{T} + \omega_n^2) + z^{-1}(2\omega_n^2 - \frac{8}{T^2}) + \frac{4}{T^2} + \zeta\omega_n\frac{4}{T} + \omega_n^2} \\
&= \frac{z^{-2} \cdot \frac{(\omega_n^2 - 2\zeta\omega_n\frac{2}{T})}{\frac{4}{T^2}+\zeta\omega_n\frac{4}{T}+\omega_n^2} + z^{-1} \cdot \frac{2\omega_n^2}{\frac{4}{T^2}+\zeta\omega_n\frac{4}{T}+\omega_n^2} + \frac{2\zeta\omega_n\frac{2}{T}+\omega_n^2}{\frac{4}{T^2}+\zeta\omega_n\frac{4}{T}+\omega_n^2}}{z^{-2} \cdot \frac{(\frac{4}{T^2} - 2\zeta\omega_n\frac{2}{T}+\omega_n^2)}{\frac{4}{T^2}+\zeta\omega_n\frac{4}{T}+\omega_n^2} + z^{-1} \cdot \frac{(2\omega_n^2 - \frac{8}{T^2})}{\frac{4}{T^2}+\zeta\omega_n\frac{4}{T}+\omega_n^2} + 1} \qquad (3.6.6)
\end{aligned}$$

Since a control system is characterized by the poles of the system, the coefficients of the denominator in equation 3.6.6 is equated to those of equation 3.6.4.

$$K_0K_pK_1 = \frac{8\zeta\omega_n T}{\omega_n^2T^2 + 4\zeta\omega_n T + 4} \tag{3.6.7}$$

$$K_0K_pK_2 = \frac{4\omega_n^2T^2}{\omega_n^2T^2 + 4\zeta\omega_n T + 4} \tag{3.6.8}$$

One could, as in the continuous time PLL design, specify the effective loop bandwidth when determining the loop constants. This is done by taking equation A.1.10 and making ω_n the subject of the equation and substituting the result into equations 3.6.7 and 3.6.8. This results in the following equations:

$$K_0K_pK_1 = \frac{8\zeta\left(\frac{2B_n}{\zeta+\frac{1}{4\zeta}}\right)T}{\left(\frac{2B_n}{\zeta+\frac{1}{4\zeta}}\right)^2T^2 + 4\zeta\left(\frac{2B_n}{\zeta+\frac{1}{4\zeta}}\right)T + 4} \tag{3.6.9}$$

$$K_0K_pK_2 = \frac{4\left(\frac{2B_n}{\zeta+\frac{1}{4\zeta}}\right)^2T^2}{\left(\frac{2B_n}{\zeta+\frac{1}{4\zeta}}\right)^2T^2 + 4\zeta\left(\frac{2B_n}{\zeta+\frac{1}{4\zeta}}\right)T + 4} \tag{3.6.10}$$

The decision of how much loop bandwidth to use in the timing recovery loop is determined by the locking range needed and the maximum allowable clock jitter. These however are conflicting requirements because for a wide locking range, the recovery loop requires a wide bandwidth which also passes a lot of noise thus increasing the clock jitter. Likewise, decreasing the loop bandwidth for less clock jitter reduces the locking range. Ideally one would want a wide as possible locking range with as little as possible clock jitter. The solution is to use a dual locking system where you use a wider bandwidth loop filter for the initial locking and then switch over to a narrower tracking filter which will minimize the clock jitter. The symbol lock detector discussed in section 3.7 determines when to switch from the pre-lock filter to the post-lock filter. The constants K1 and K2 are used for the pre-locking filter and K3 and K4 for the post-lock filter. When the lock detector detects the locked condition, integrator 2 adds the integral component of the pre-lock filter output before resetting integrator 1. This preserves the steady state frequency of the pre-lock filter. The coefficients of the post lock filter are then selected.

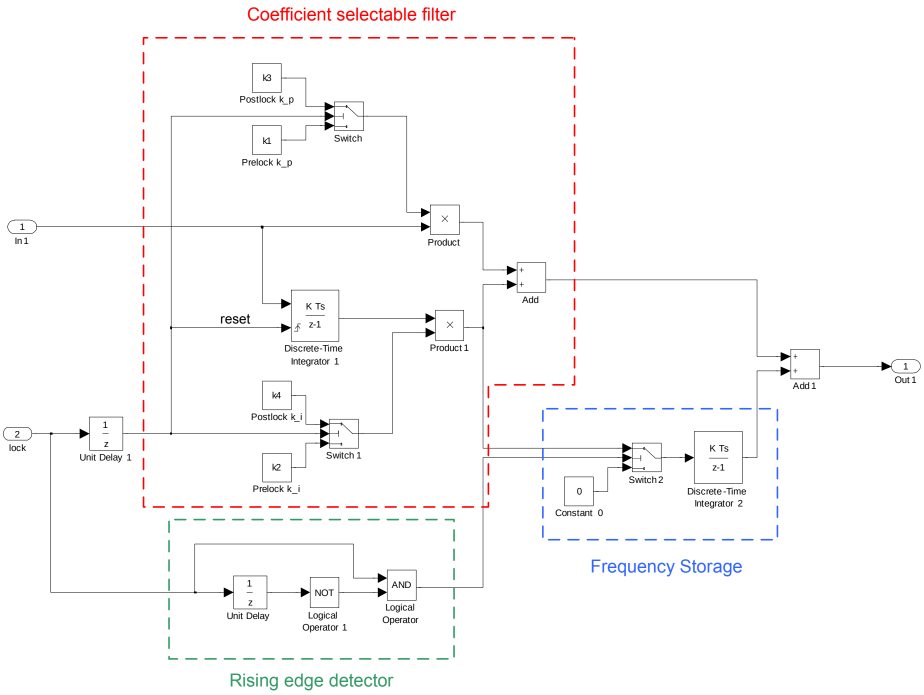


Figure 3.29: Dual lock timing loop filter

The latency of the ADC and DDS are the main limiting factors influencing the maximum loop bandwidth. The various blocks in the symbol recovery loop all have latencies that are specified at the clock frequency of the respective blocks which is not necessarily the symbol clock rate. These latencies need to be expressed as multiples of the symbol rate which is the update rate of the recovery loop. The ADC has a latency of ten cycles running at 100 MHz which is only five cycles at 50 MHz. The DDS latency is 79 cycles running at 1GHz which results in four cycles at 50 MHz. There is also the programming delay of the DDS. The parallel programming interface has a four cycle delay. The total loop delay for the parallel interface is 18 cycles. Simulations in section 4.1 shows that a bandwidth of 25 kHz for the pre-lock filter can be achieved with negligible effects from the transport delays. The bandwidth of the post-lock filter is set to 2.5 kHz.

3.7 Symbol lock detection

This section discusses the lock detection algorithm used to determine when the timing recovery loop is locked.

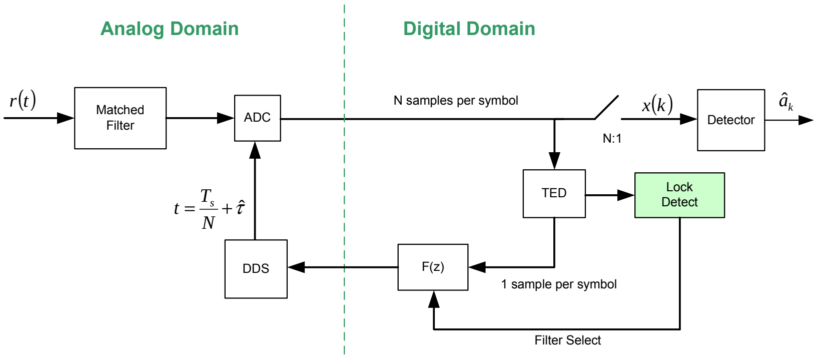


Figure 3.30: Synchronous baseband demodulation with DDS clock

In [6], a symbol lock detector is proposed which operates on two samples per symbol and is thus suitable for this QPSK demodulator. The sampled data needs to be separated into even and odd samples which correspond to the peak and zero crossings of the sampled matched filter output when in the locked state. The lock detect algorithm is shown in equation 3.7.1.

$$S_N = \frac{1}{N}\sum_{k=1}^{N} \frac{I_e(k)^2 - I_o(k)^2}{I_e(k)^2 + I_o(k)^2} + \frac{Q_e(k)^2 - Q_o(k)^2}{Q_e(k)^2 + Q_o(k)^2} \tag{3.7.1}$$

Two important parameters of lock detectors are their detection rate (P_D) and the false alarm rate (P_{FA}). The detection rate is the probability of detecting the lock condition given that the system is locked. The false alarm rate is the probability of detecting a lock condition given that the system is not locked.

Given a P_D, P_{FA} and the $\frac{E_s}{N_0}$ at which the two preceding probabilities are specified, the number of accumulated samples needed to satisfy the detection probabilities is calculated with equation 3.7.2.

$$N = 8\left(\frac{erfc^{-1}(2P_{FA}) - erfc^{-1}(2P_D)}{\mu_s(\frac{E_s}{N_0})}\right)^2 \tag{3.7.2}$$

$$\mu(\frac{E_s}{N_0}) \approx \exp\left(-2\frac{N_0}{E_s}\right) \tag{3.7.3}$$

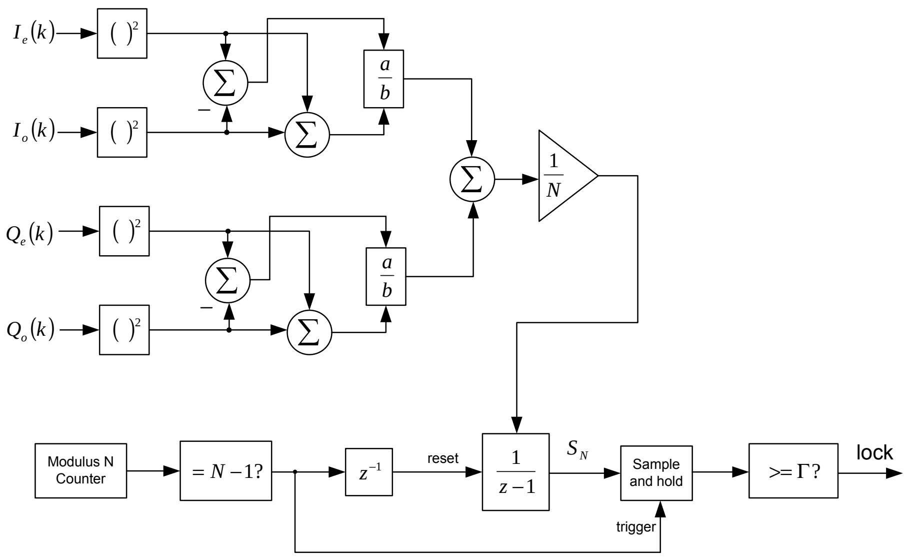


Figure 3.31: Symbol lock detector

The function $\mu(\frac{E_s}{N_O})$ is an approximation of the expected value of S_N when the symbol recovery loop is locked. The threshold level with which the detector decides between the locked and unlocked state is calculated with 3.7.4.

$$\Gamma = \frac{erfc^{-1}(2P_{FA}) \cdot \mu_s(\frac{E_s}{N_0})}{erfc^{-1}(2P_{FA}) - erfc^{-1}(2P_D)} \tag{3.7.4}$$

To illustrate the relationship between the P_D, P_{FA} and the number of samples needed, the probability of detection is plotted against the number of samples for various false detection rates in figure 3.32. From figure 3.32, it is clear that the more samples that are used for the lock detect decision, the higher the probability of detecting the locked condition. The $P_{FA} = 10^{-12}$ trace shows for a $P_D = 0.999$, the number of 2N samples required are calculated as 611. The number of samples selected is rounded up to the next power of two which 1024. Figure 3.33 illustrates the how the probability of detection changes for different decision thresholds. A detection threshold of 0.57 produces a $P_D = 0.999$ for a $P_{FA} = 10^{-12}$.

This lock detector has the added feature that the signal to noise ratio can be fairly accurately estimated with equation 3.7.5.

$$\left(\frac{E_s}{N_0}\right)_{dB} \approx 10\log_{10}\left(\frac{-2}{\ln(S_n)}\right) \tag{3.7.5}$$

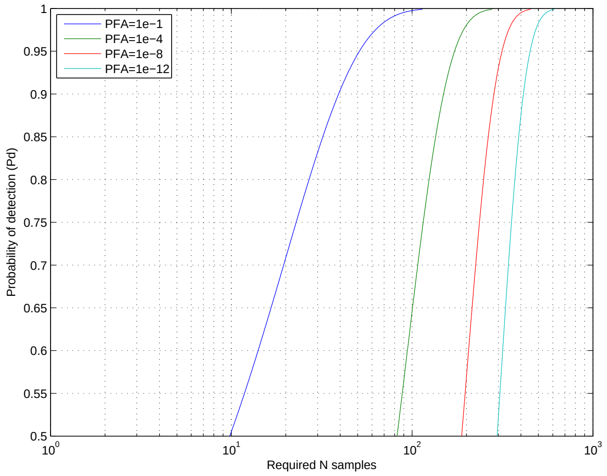


Figure 3.32: Required number of samples needed to satisfy P_D and P_{FA} with $\frac{E_S}{N_0} = 10dB$

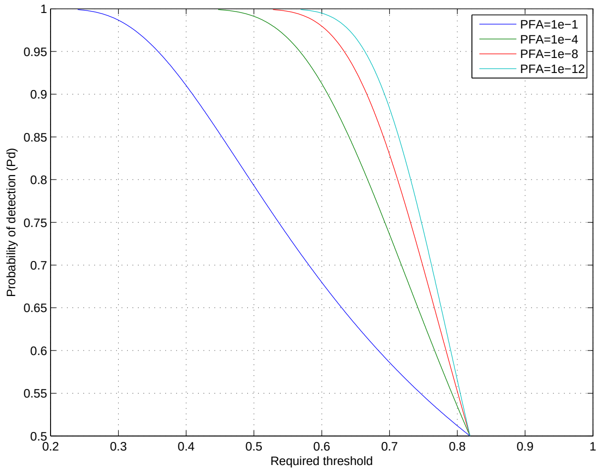


Figure 3.33: Required threshold (Γ) needed to satisfy P_D and P_{FA} with $\frac{E_S}{N_0} = 10dB$

3.8 Detector

The detector, also known as the decision device, identifies the received baseband symbols. The detector is just a comparator that checks if the received baseband signal exceeds a preset threshold and thereby identifying the symbol.

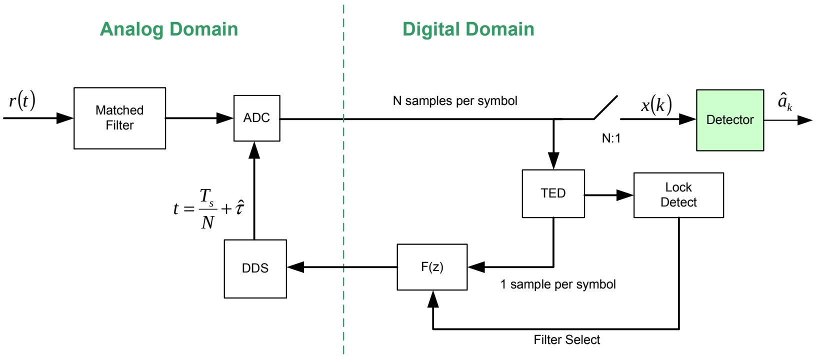


Figure 3.34: Synchronous baseband demodulation with DDS clock

The noise at the input of the matched filter is zero mean white Gaussian noise which produces a noise spectral density at the output of the matched filter given by equation 3.8.1[4].

$$S_{n0}(f) = \frac{1}{2} N_0 |G_m(f)|^2 \tag{3.8.1}$$

$S_{n0}(f)$ is integrated over frequency to get the variance or noise power at the output of the matched filter.

$$\sigma_0^2 = \int_{-\infty}^{\infty} \frac{1}{2} N_0 |G_m(f)|^2 df \tag{3.8.2}$$

The noise probability density function is given by equation 3.8.3.

$$f_N(\eta) = \frac{e^{-\eta^2/2\sigma_0^2}}{\sqrt{2\pi\sigma_0^2}} \tag{3.8.3}$$

When the $s_1(t)$ is transmitted, the peak sampled value of the matched filter is

$$X \triangleq x(T_s) = s_{01}(T_s) + N \tag{3.8.4}$$

Likewise, when $s_2(t)$ is transmitted the matched filter sampled value is

$$X \triangleq x(T_s) = s_{02}(T_s) + N \tag{3.8.5}$$

The pdf of the peak matched filter samples has a Gaussian random variable with a variance of σ_0^2 and mean $s_{01}(T_s)$ or $s_{02}(T_s)$ depending on which symbol was sent.

$$f_X(x/s_1(t)) = \frac{e^{-(x-s_{01}(T_s))^2/2\sigma_0^2}}{\sqrt{2\pi\sigma_0^2}} \tag{3.8.6}$$

The conditional probability density functions when $s_1(t)$ or $s_2(t)$ is received are plotted in figure 3.35 with a deliberate DC offset. The section where the pdf's overlap is the region where the noise corrupts the received signal to the extent where the wrong symbol is more likely to be identified.

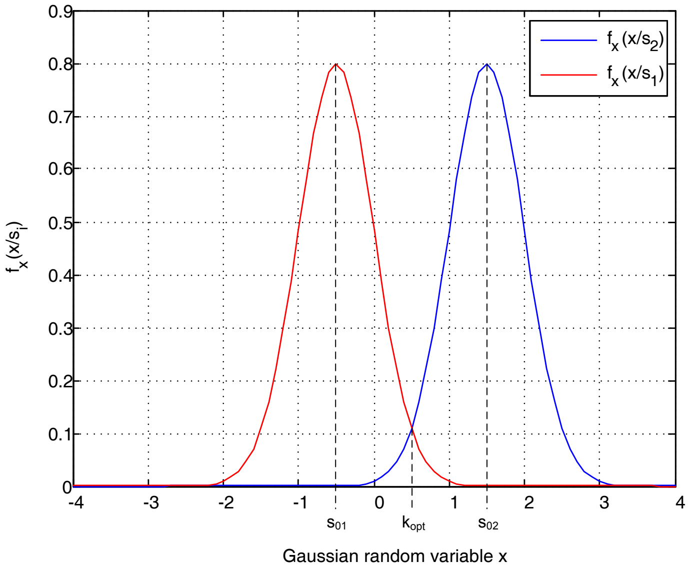


Figure 3.35: The conditional probability density function of the peak matched filter output.

The probability of error given s_1 was sent is the area under $f_x(x/s_1)$ to the right of $x = k$.

$$\begin{aligned} P(E/s_1) &= \int_k^\infty f_X(x/s_1(t))dv \qquad (3.8.7)\\ &= \int_k^\infty \frac{e^{-(x-s_{01}(T_s))^2/2\sigma_0^2}}{\sqrt{2\pi\sigma_0^2}} \qquad (3.8.8)\end{aligned}$$

Similarly, the $P(E/s_2(t))$ is calculated with equation 3.8.9.

$$P(E/s_2) = \int_{-\infty}^k \frac{e^{-(x-s_{02}(T_s))^2/2\sigma_0^2}}{\sqrt{2\pi\sigma_0^2}} \tag{3.8.9}$$

The *a priori* probabilities of s_1 and s_2 are equal which results in an average probability of error of

$$P_E = \frac{1}{2}P(E/s_1(t)) + \frac{1}{2}P(E/s_2(t)) \tag{3.8.10}$$

The optimum threshold k for the comparator is easily shown to be

$$k_{opt} = \frac{1}{2}\left[s_{01}(T_s) + s_{02}(T_s)\right] \tag{3.8.11}$$

when the *a priori* probabilities of the received symbols are equal. The optimum threshold is the DC average of the matched filter output. In section 6.3.2 a DC compensation block is designed that removes any DC offset from the matched filter output which means that the threshold k is simply set to zero.

3.9 Conclusion

In this chapter, a binary PAM signal was mathematically described and the unknown parameters, pertaining to the reception of such a PAM signal, were identified. A strategy was proposed to estimate the phase delay of a received baseband signal by implementing synchronous sampling. The synchronous sampling is implemented by using a DDS to generate the variable clock for the ADC. A matched filter matched to the received rectangular pulses was designed and implemented with discrete components because the ADC sample rate is only twice the symbol rate and does not satisfy Nyquist's sampling theorem. An early late timing error detector was designed which extracts timing information from the received pulses. The timing recovery loop was modeled as a linear second order system. The loop filter coefficients were mathematically derived for an arbitrary damping factor and loop bandwidth. A timing lock detector was designed to switch from a wide band pre-locking filter, to a narrow band post-lock filter. The result is a recovery loop with wide locking range and low jitter. The decision device, designed for a binary PAM signal, is simply a zero threshold level comparator.

Chapter 4

Baseband signal recovery simulation

4.1 Baseband signal recovery simulation overview

The simulation of the baseband symbol recovery algorithm in Matlab Simulink is one of the bigger milestones of the research that was done. The problem comes in that there is a transmitter generating data at one bit rate and also a receiver whose sample rate changes to lock the receiver to the transmitter data rate. At first glance one would try and implement the simulation exactly as the system operates by keeping the transmitter data rate fixed and varying the clock rate of the receiver logic. This implementation was tried with fair degree of success. The real problems started to present themselves when the Simulink DSP Builder blocks from Altera were integrated. DSP Builder does not provide for variable sampling rates of its blocks so the solution was to make them fixed and control the data rate of the transmitter. The discrete sampling rate of the simulation is 6.4 GHz. The symbol rate of the demodulator is 50MSPS which means the simulation has a time resolution of 128 samples per symbol. An important parameter that needed to be examined was the loop gain and the factors scaling it.

The Simulink simulation model for the timing recovery loop is shown in figure 4.1. It consists of a transmitter, channel, receiver and a bit error rate calculator. The transmitter can generate a random bipolar signal or an alternating pulse sequence which is filtered with a 600 MHz Bessel filter. The channel adds white Gaussian noise. At the receiver, the received signal is sent through a matched filter. Then the filtered signal is sampled and quantized at a rate of two samples per symbol. The sampled symbols go to the timing error detector which has three selectable implementations for the TED namely the Early late TED, Zero crossing TED and the Gardner TED. Repetitive simulations are done to evaluate the performance of the different timing error detectors. The timing error signal goes through the loop filter into the DDS clock block which simulates the parallel programming interface of the DDS. The DDS clock is used to clock the transmitter and bit error rate calculator.

First a simulation is done with alternating data to illustrate the basic operation of the simulation model with the early late TED enabled. The receiver data rate is set exactly the same as the trans-

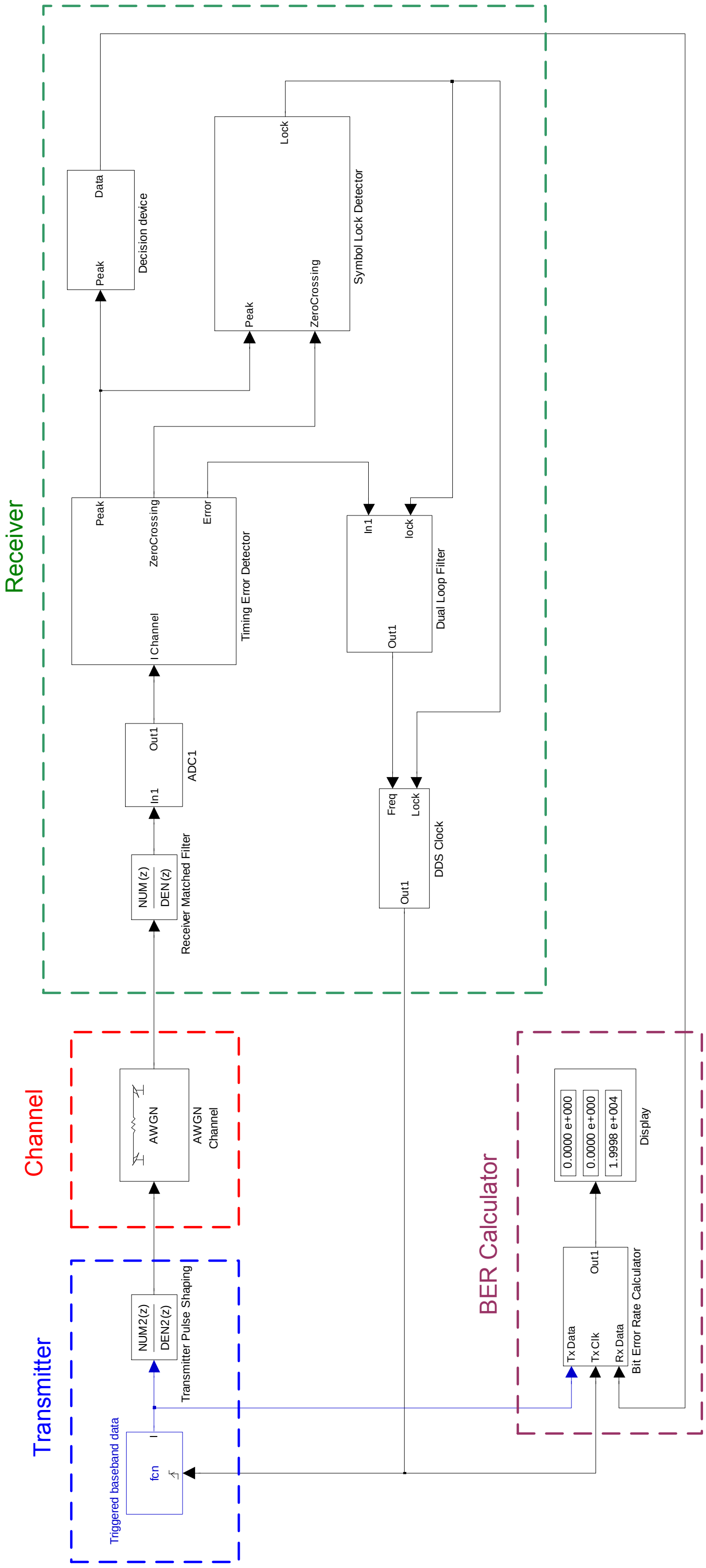


Figure 4.1: Baseband timing recovery Simulink model.

mitters data rate but with a pi/2 initial phase shift is. The equivalent noise bandwidth (B_L) is set to 25 kHz and the post locking bandwidth is 2.5 kHz. Non-ideal effects such as quantization noise and hardware delays are first ignored.

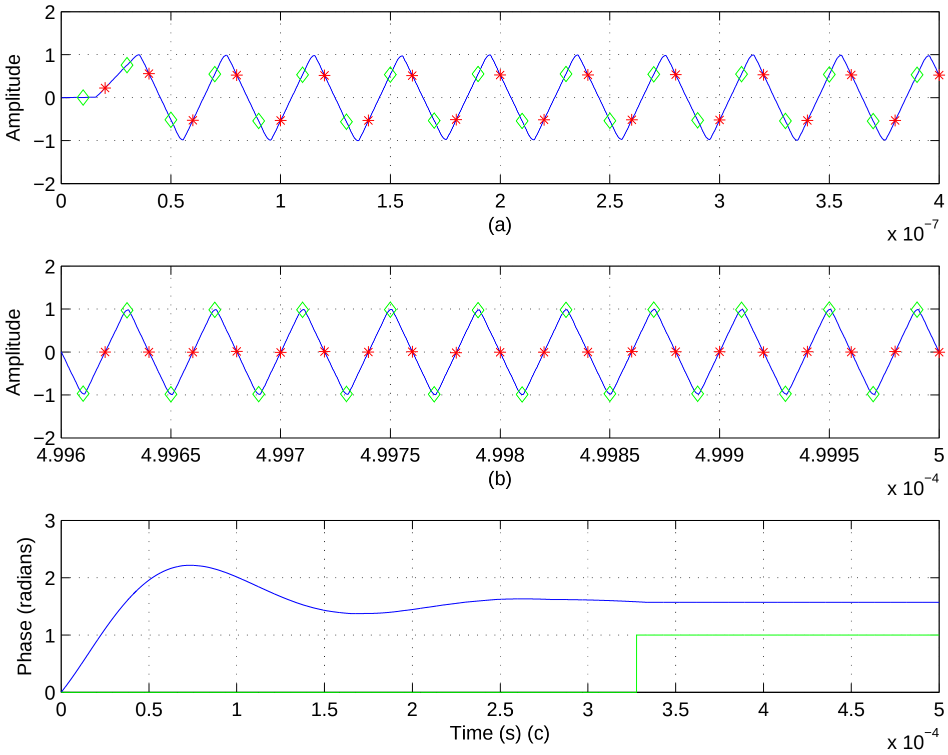


Figure 4.2: Baseband recovery simulation: (a) Pre-lock baseband sampling. (b) Post-lock baseband sampling. (c) Timing recovery loop phase step response.

The red stars in figure 4.2(a,b) are the estimated symbol transition positions and the green diamonds are the estimated middle positions of the baseband symbols. In figure 4.2(a), the recovery loop is not locked and thus the estimated transitions are not on the actual symbol transitions. In figure 4.2(b), the system has locked with the red stars on the symbol transitions and the green diamonds in the middle of the symbols. Figure 4.2(c) shows the phase step response of the recovery loop. The step response introduces a phase step beyond the linear region of the phase detector. To extract the loop parameters of the recovery loop, a simulation with smaller phase step is done.

The phase step response for the early late TED, zero crossing TED and Gardner TED are shown in figure 4.3. The phase step is $\frac{\pi}{16}$ to operate in the more linear region of the phase detectors. The response of the early late TED and the Gardner TED lie almost perfectly on top of each other with a significantly higher overshoot that the Zero crossing TED.

From table 4.1 one can see that the loop bandwidth is quite close to the designed 25 kHz. The

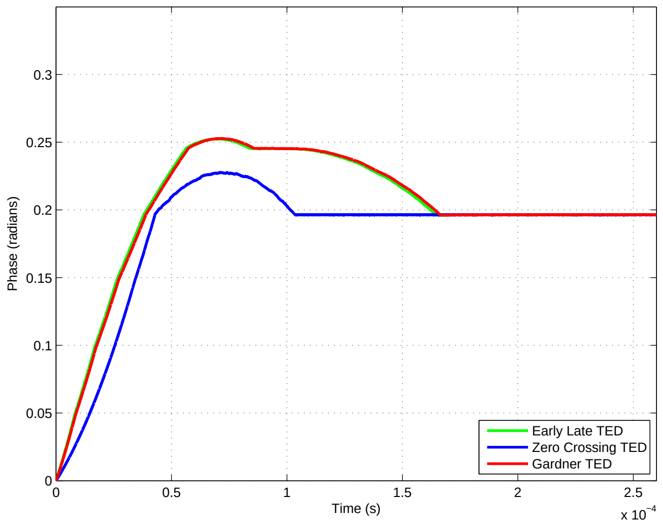


Figure 4.3: Ideal baseband recovery simulation with three different TED's with alternating data

Type	B_n	ζ
EL-TED	25.5 kHz	0.37
ZC-TED	25.6 kHz	0.5
Gardner TED	24.8 kHz	0.37

Table 4.1: Ideal symbol recovery loop simulation results for alternating data

damping factor however shows significant overshoot when taking into account that a damping factor of 0.707 was specified. When examining the waveforms of the step responses one sees that the response has non linear characteristics and the original linearized second order model is just an approximation.

The QPSK demodulator will operate on random data and not alternating data and thus there will not be a transition after every bit. The loop gain halves when the data source is changed to a random binary data source which generates symbols with equal *a priori* probabilities. The random data produce halve the amount of transitions when compared to an alternating binary source.

The steady state level of the TED step responses show that the TED's each lock with a slightly different phase offset. The offsets are very small and will have no noticeable impact on the performance of the demodulator. When examining figure 4.4 and table 4.2 the loop bandwidths stayed reasonably constant when random data is used as the data source except for the Zero crossing TED who's bandwidth reduced by slightly more than 20%. The Zero crossing TED has the most non linear operation of the three TED's but it has lower self noise than the other two TED's. Self noise is noise that is generated by a TED in the absence of transitions. Ideally the

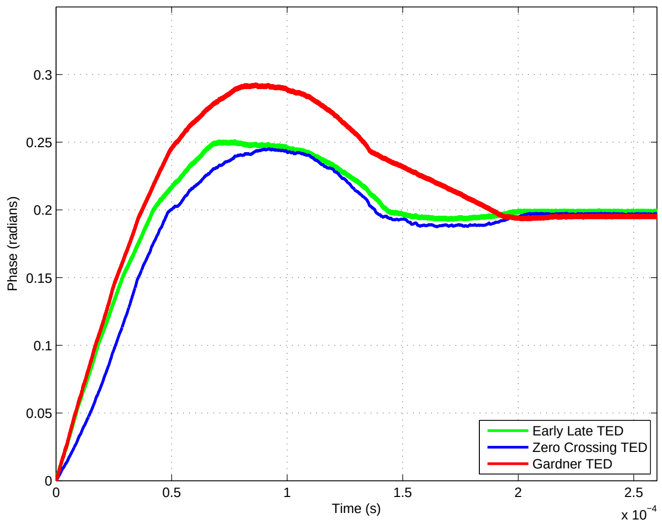


Figure 4.4: Ideal baseband recovery simulation with three different TED's with random data

Type	B_n	ζ
EL-TED	22.8 kHz	0.38
ZC-TED	19 kHz	0.4
Gardner TED	25.3 kHz	0.22

Table 4.2: Ideal symbol timing recovery loop simulation results for random data

output of the TED should be zero when there are no transitions. Examining figure 4.5, one can easily see that the self noise of the Early late TED and Gardner TED is worse than the self noise of the Zero crossing TED.

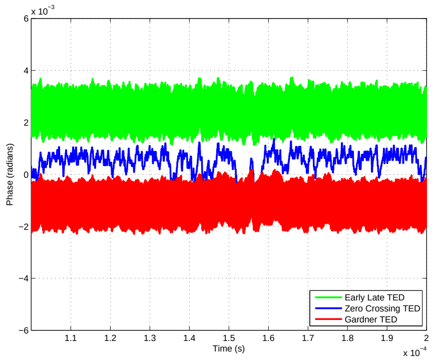


Figure 4.5: Self noise of three different TED's with random data

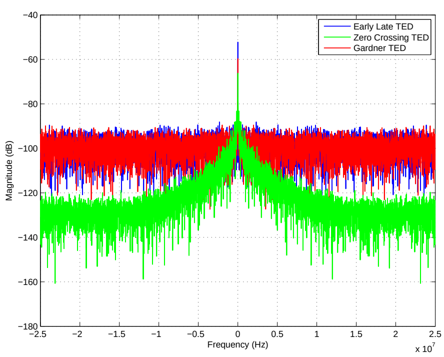


Figure 4.6: Frequency spectrum of TED's output with random data source

Table 4.3 shows that the self noise of the Zero crossing TED is about four times less than the other two TED's. The delays through the recovery loop of the DDS have not been included in

Type	$\sigma_\phi^2(25kHz)$	$\sigma_\phi^2(2.5kHz)$
EL-TED	2.8e-3	2.8e-5
ZC-TED	6.1e-4	7.6e-6
Gardner TED	2.7e-3	2.7e-5

Table 4.3: Self noise generated by TED's from random data

the simulation model and has a significant impact on the performance of the timing recovery. The delays through the recovery loop consist of pipeline delays and zero order hold delays. The zero order hold delays are primarily caused by the programming process. While the DDS is being programmed, any new frequency information sent to the programming block are ignored which basically results in down sampling. In figure 4.7 step responses of the timing recovery loops with alternating data seem unaffected by the delays and down sampling, but when the data source is changed to random data, the recovery loops with the early late TED and Gardner TED become unstable. The step response of the zero crossing TED however has only a slight increase in self noise. When a windowed integrator is inserted after the loop filter to average the frequency signal over the time that DDS is programmed, one can slightly improve the phase estimate by not throwing the samples away when programming is taking place. The zero crossing detector clearly has much less self noise for this hardware implementation where programming delays are present. The pre-locking bandwidth for the zero crossing TED implementation is 19.7 kHz.

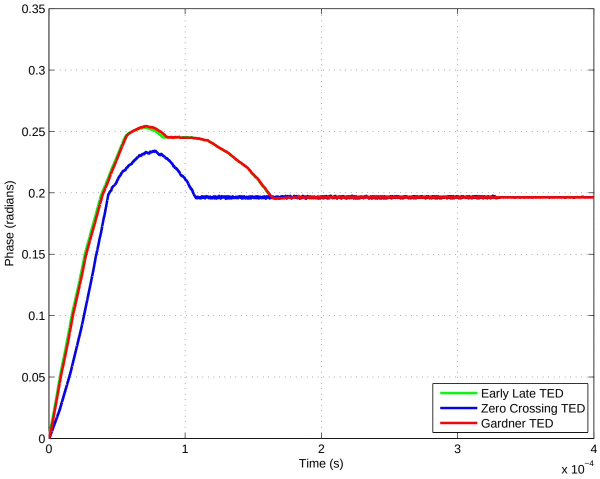


Figure 4.7: Step response of timing recovery loop with loop delays for alternating data

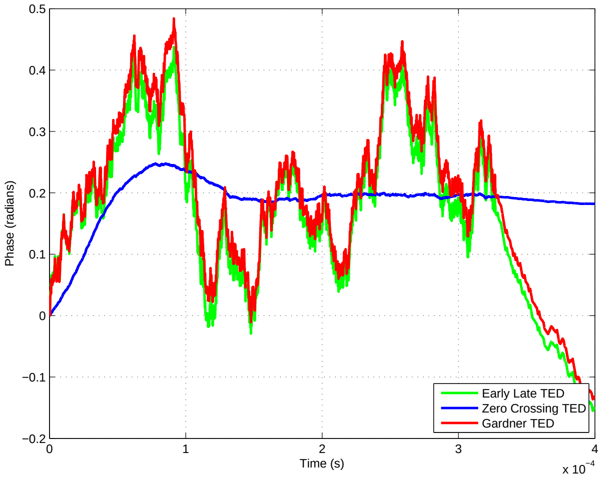


Figure 4.8: Step response of timing recovery loop with loop delays for random data

The green trace in figure 4.10 is the BER theoretical limit for a bipolar binary signal. The open loop and rest of the closed loop simulations with the various TED's all fall almost exactly on top of each other. The difference in performance compared to the theoretical limit is about 0.2

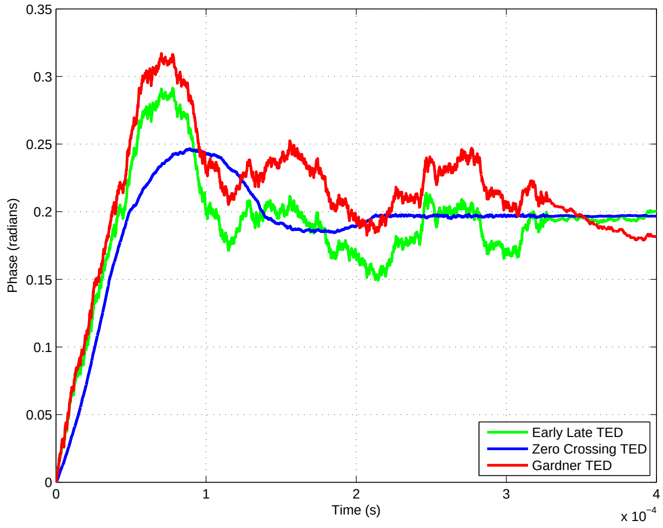


Figure 4.9: Timing step responses with programming delay compensation

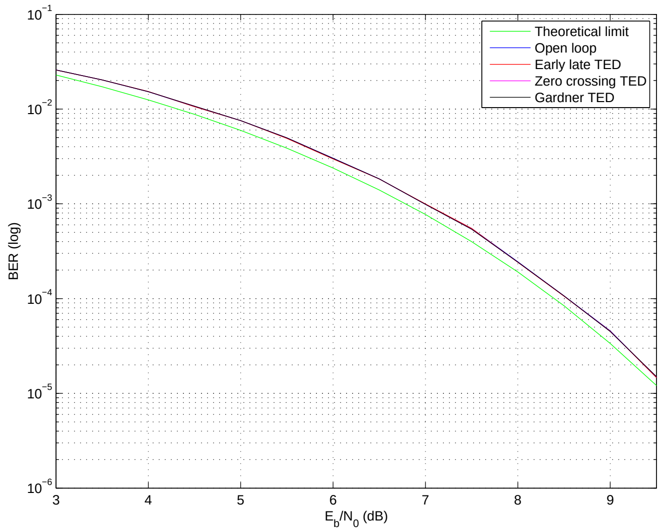


Figure 4.10: Bit error rate for baseband symbol recovery without delays

dB with means that the E_b/N_0 must be 0.2 dB higher for the same BER when compared to the theoretical limit. The implementation of the matched filter is the main reason for the difference in performance.

Chapter 5

QPSK modulation

5.1 QPSK modulation overview

This chapter discusses the modulation process for a QPSK signal. The modulator consists of the FPGA development board with two on-board DAC's and an I/Q modulator board. The modulation hardware is shown in figure 5.1.

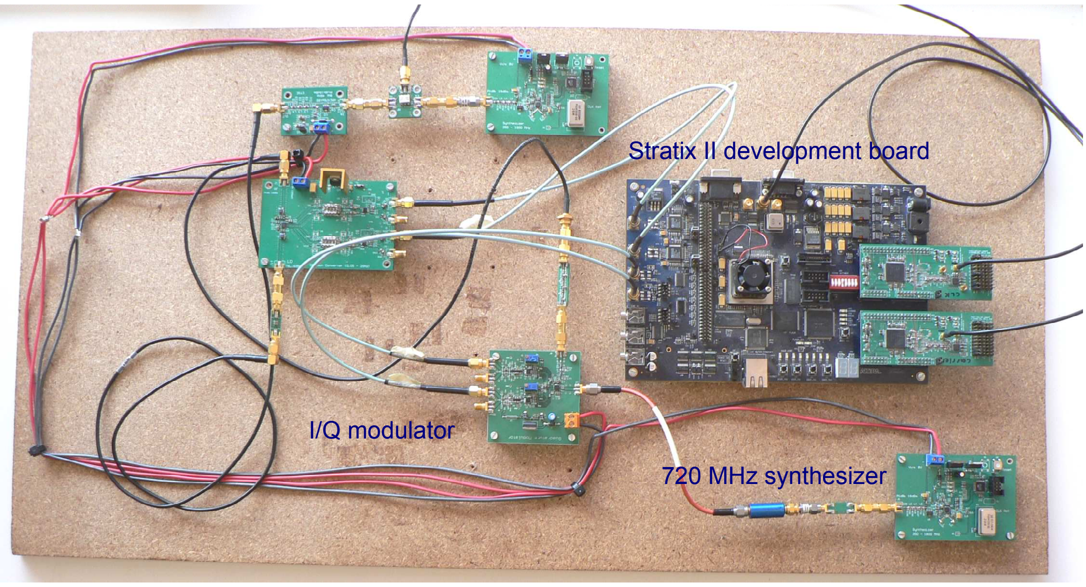


Figure 5.1: QPSK modulator hardware.

First, the theory involved in the modulation process will be shown and how it is represented in block diagram form. Then the hardware is discussed followed by simulations showing the effects of imperfections in the modulation process.

5.2 Passband PAM signal representation

The modulation process for a QPSK signal is mathematically almost identical to QAM modulation which is also passband PAM modulation. A passband PAM signal is expressed in equation 5.2.1.

$$s_{IF}(t) = \Re\{s_{CE}(t)e^{j2\pi f_c t}\} \tag{5.2.1}$$

s_{CE} is the signal complex envelope relative to the carrier frequency f_c. The complex envelope has the same form as the baseband signal representation in equation 3.1.1 except c_i can be complex. T_s is the symbol period.

$$s_{CE}(t) = \sum_i c_i g(t - iT_s) \tag{5.2.2}$$

For a M-ary PSK signal, c_i is a complex exponential with $\alpha_i \in \{0, 2\pi/M, ..., 2\pi/(M-1)\}$ and could be expanded into real and imaginary components using Euler's rule.

$$c_i = e^{j\alpha_i} = a_i + jb_i \tag{5.2.3}$$

To implement passband PAM modulation, equations 5.2.2 and 5.2.1 are expanded.

$$s_{IF}(t) = \cos(2\pi f_c t)\sum_i a_i g(t - iT_s) - \sin(2\pi f_c t)\sum_i b_i g(t - iT_s) \tag{5.2.4}$$

Figure 5.2 shows the block diagram of a quadrature modulator. In a QPSK receiver, the carrier recovery block can lock onto any one of the four symbol phases that were transmitted. This introduces a $\pi/2$ phase ambiguity in the receiver which results in the constellation being rotated by $\pi/2$,$-\pi/2$ or π. To solve this problem, there are a few strategies that could be followed. The first method is to send a known preamble code which the receiver can use to correct the

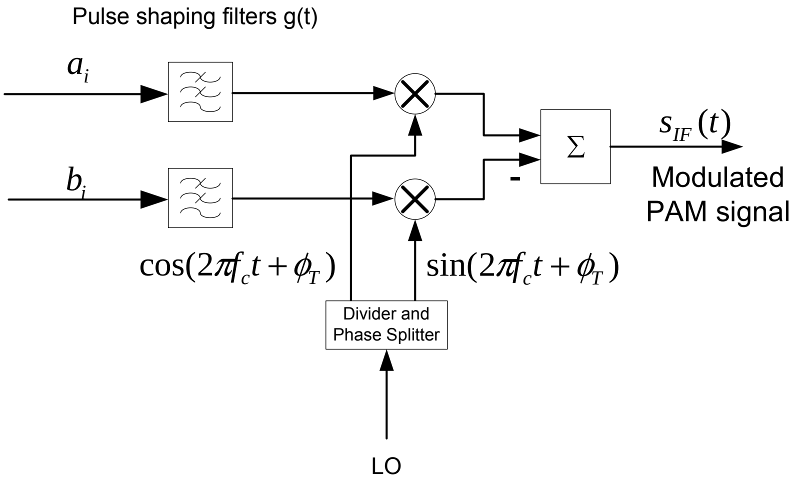

Pulse shaping filters g(t)
a_i
b_i
Σ
-
$s_{IF}(t)$
Modulated PAM signal
$\cos(2\pi f_c t + \phi_T)$
$\sin(2\pi f_c t + \phi_T)$
Divider and Phase Splitter
LO

Figure 5.2: Passband PAM signal modulator

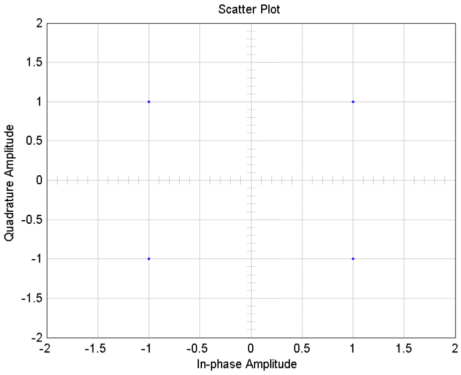


Figure 5.3: QPSK scatter diagram

phase ambiguity. The second method is to differentially encode the data which means that the information is not contained in the absolute phase of the transmitted symbols, but in the phase difference from the previous symbol. The drawback of differential encoding is that if a symbol is erroneously identified in the receiver, the next symbol will also be wrong. This is due to the relative phase jump from the previous wrong symbol that will be incorrectly calculated. To generate differentially encoded data, the lookup table in table 5.1 was used.

A(k)	B(k)	I(k-1)	Q(k-1)	I(k)	Q(k)	Delta Phase
0	0	0	0	0	0	0°
0	0	0	1	0	1	0°
0	0	1	1	1	1	0°
0	0	1	0	1	0	0°
0	1	0	0	1	0	90°
0	1	0	1	0	0	90°
0	1	1	1	0	1	90°
0	1	1	0	1	1	90°
1	1	0	0	1	1	180°
1	1	0	1	1	0	180°
1	1	1	1	0	0	180°
1	1	1	0	0	1	180°
1	0	0	0	0	1	-90°
1	0	0	1	1	1	-90°
1	0	1	1	1	0	-90°
1	0	1	0	0	0	-90°

Table 5.1: Differential encoding lookup table

Another method to resolve the phase ambiguity requires that some form of error detection be implemented. When the demodulator detects a lock condition but the error detection logic measures an excessive amount of errors, it assumes that the constellation is rotated and a $\frac{\pi}{2}$ rotation is done. The symbol error rate is checked again and the same action is again taken if the symbol error rate is too high.

5.3 Non ideal effects in QPSK modulation

This section was included to give a practical understanding into the factors influencing the modulation process. As mentioned earlier, a common performance benchmark used for quadrature modulation is the single sideband (SSB) suppression. Assume for a moment that the baseband data signal of the in-phase and quadrature channels are sinusoidal and orthogonal. The QPSK modulator, with perfect orthogonal internal carriers, produces an output signal as described in equation 5.3.1

$$\begin{aligned}
s_{IF}(t) &= \cos(2\pi f_c t)\cdot a\cos(2\pi f_b t) - \sin(2\pi f_c t)\cdot b\sin(2\pi f_b t) \\
&= \cos(2\pi f_c t)\cdot a\cos(2\pi f_b t) - \sin(2\pi f_c t)\cdot a\sin(2\pi f_b t) \\
&\quad - \sin(2\pi f_c t)\cdot (b-a)\sin(2\pi f_b t) \\
&= a\cos(2\pi(f_c+f_b)t) - (b-a)/2\cdot\cos(2\pi(f_c-f_b)t) + (b-a)/2\cdot\cos(2\pi(f_c+f_b)t) \\
&= (a+b)/2\cdot\cos(2\pi(f_c+f_b t)) + (a-b)/2\cdot\cos(2\pi(f_c-f_b)t) \qquad (5.3.1)
\end{aligned}$$

In equation 5.3.1 the parameters 'a' and 'b' are the baseband amplitudes of the in-phase and quadrature channels. Two sidebands are generated in the quadrature modulator. One can see that when the baseband amplitudes are equal, the second term in equation 5.3.1 is zero and only the upper sideband is generated. When they are not equal, the lower sideband is a function of the difference in baseband amplitude. The single sideband rejection, which is typically quoted in product data sheets, refers to the rejection of the one sideband relative to the other. Another parameter that affects the single sideband rejection, is the in-phase and quadrature carrier orthogonality. Let the phase difference between the two carriers be $\pi/2+\theta$. With this phase offset, equation 5.3.1 changes to equation 5.3.2

$$\begin{aligned}
s_{IF}(t) &= \cos(2\pi f_c t)\cdot a\cos(2\pi f_b t) - \sin(2\pi f_c t+\theta)\cdot b\sin(2\pi f_b t) \\
&= a/2\cos(2\pi(f_c+f_b)t) + a/2\cos(2\pi(f_c-f_b)t) + b/2\cos(2\pi(f_c+f_b)t+\theta) \\
&\quad -b/2\cos(2\pi(f_c-f_b)t+\theta) \\
&= \sqrt{\frac{a^2}{4}+\frac{ab}{2}\cos(\theta)+\frac{b^2}{4}}\cdot\cos\left(2\pi(f_c+f_b)t+\arctan\left(\frac{\frac{-b}{2}\sin(\theta)}{\frac{a}{2}+\frac{b}{2}\cos(\theta)}\right)\right) + \\
&\quad \sqrt{\frac{a^2}{4}-\frac{ab}{2}\cos(\theta)+\frac{b^2}{4}}\cdot\cos\left(2\pi(f_c-f_b)t+\arctan\left(\frac{\frac{b}{2}\sin(\theta)}{\frac{a}{2}+\frac{b}{2}\cos(\theta)}\right)\right) \qquad (5.3.2)
\end{aligned}$$

The baseband amplitude imbalance is typically not a problem since the baseband amplitudes can be set externally to the quadrature modulator. The phase imbalance of the quadrature carriers however are internally generated and can not be corrected. The single sideband rejection ratio can also be measured by using square baseband signals. This is done by generating alternating baseband data that is shifted by half a bit period. Only the fundamental components of the in-phase and quadrature baseband signals will be orthogonal. From equation 5.3.2, the single sideband rejection for a phase imbalance of 5° is 41.2 dB. Two simulations were done to illustrate the effects of a 5° phase imbalance. First a 1 kHz carrier is modulated with a 10 Hz sinusoidal tone. Then a second simulation is done to illustrate single sideband generation with a square wave baseband signal. The results of these simulations are shown in figures 5.4 and 5.5 respectively. In figure 5.4, the sideband rejection is easily verified as 41.2 dB since there are only two sidebands. For the SSB generation with square wave baseband signals, there are multiple upper and lower sidebands. Only the fundamental component of the baseband data signals are orthogonal. The ratio of the first upper and lower sideband also produces a rejection ratio of 41.2 dB. The SSB generation with baseband square waves is an easy measurement to implement for the modulation system, since no extra signal sources are needed. Only a small modification to the FPGA logic is needed to delay the one channel by half a bit period. The data source of the modulator is used as the baseband source which has the added advantage that the measurement includes any amplitude imbalance from the baseband data source used for the modulator as well.

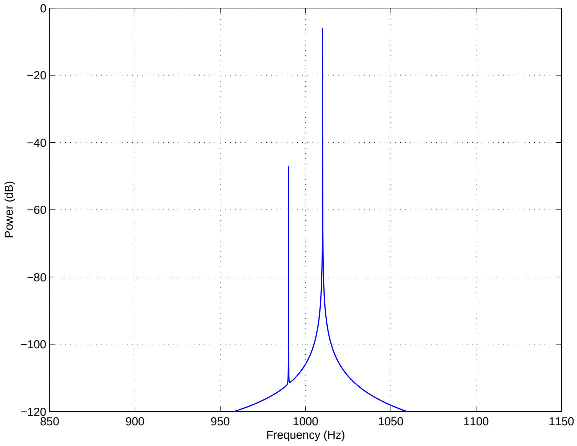


Figure 5.4: Single sideband generation with 5°carrier phase imbalance and a sinusoidal baseband signal.

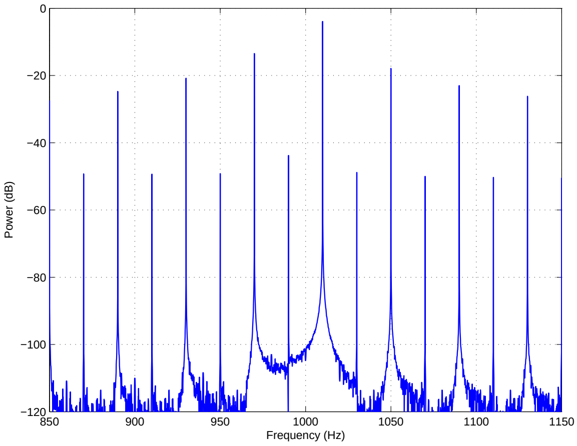


Figure 5.5: Single sideband generation with 5°carrier phase imbalance and a square wave baseband signal.

5.4 QPSK modulator hardware overview

The QPSK modulator consists of a FPGA development board, a quadrature modulator board and a synthesizer. The QPSK modulator is shown in figure 5.6. The Stratix II FPGA development board has two 160 MS/s digital to analog converters on board which are clocked at 50 MHz to generate the baseband data. The modulator logic driving the DAC's can implement any constellation up to 256 points. This is done by using a lookup table for each information symbol and mapping that symbol to a position on the constellation diagram. The clock for the DAC's and encoding logic in the FPGA is generated by a crystal oscillator. The FPGA differentially encodes the data and generates the in-phase and quadrature data streams which are used to modulate the carrier. The output of the DAC's are amplified and level shifted before it is sent to the quadrature modulator. There is also an IF filter on the output of the quadrature modulator to suppress any out of band spurious components generated by the modulation process. A synthesizer is used to generate a stable 720 MHz carrier for the IQ modulator. The output of the synthesizer is buffered with an amplifier to amplify and isolate the synthesizer output.

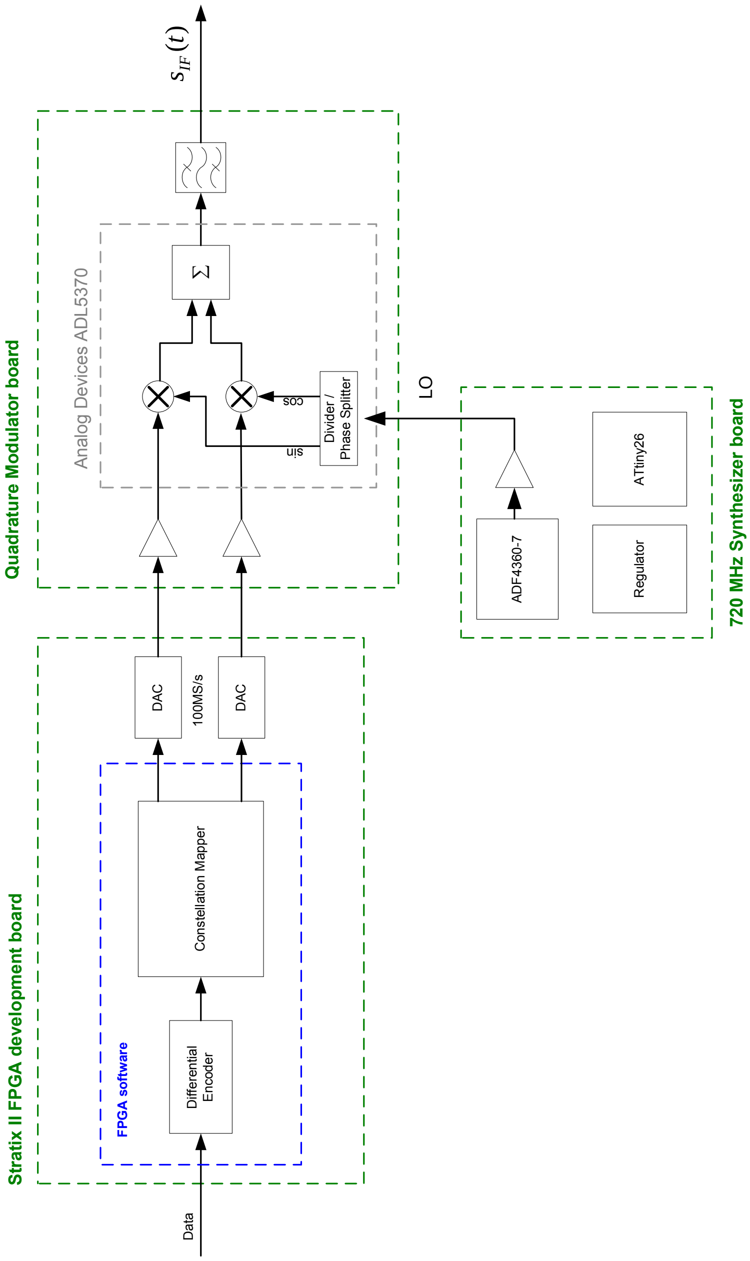


Figure 5.6: Passband PAM signal modulator block diagram

5.5 I/Q modulator

The purpose of the I/Q modulator is to translate the complex baseband signal to the intermediate frequency of 720 MHz. The modulator IC selected to perform this function was the ADL5370 from Analog Devices. This modulator was selected because it has a high baseband bandwidth (500 MHz) and good single sideband suppression (-40 dB). There are two main factors that determine the single sideband suppression of a quadrature modulator. One is the orthogonality of the in-phase and quadrature carriers generated inside the modulator IC. The other factor is the in-phase and quadrature baseband signal amplitude and phase imbalance at the mixing blocks.

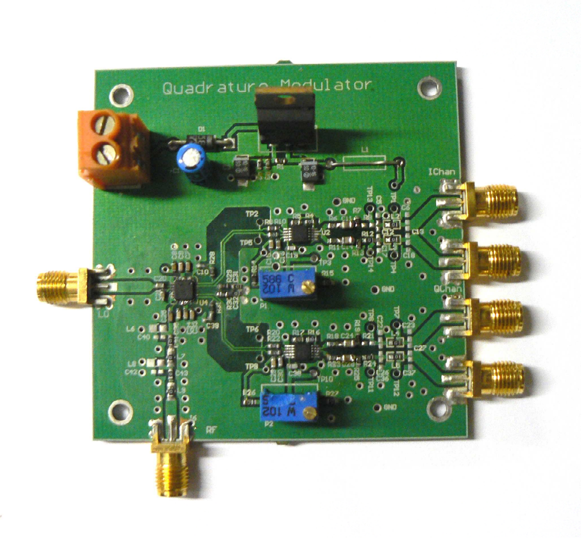


Figure 5.7: I/Q modulator board.

The baseband inputs to the IQ modulator are differential ports. Differential amplifiers are used to amplify and level shift the outputs of the digital to analog converters. The differential amplifiers are the AD8351 also from Analog Devices. The amplifier gain is set to 6 dB which produces a differential voltage swing of 1.4 V and also gives an common DC voltage offset of 0.5 V. Differential transmission lines were used to connect the operational amplifiers to the baseband inputs of the modulator. A differential characteristic impedance of 150 Ω was selected for the transmission lines.

Great care was taken to ensure that the track lengths of the differential pairs were equal to maximize the single sideband suppression of the quadrature modulator. The output of the quadrature modulator contains baseband signals that leaks through the quadrature mixer which is removed by the bandpass filter on the output. The filter is a third order Bessel filter centered at 720 MHz with a bandwidth of 240 MHz. 50 Ω coplanar transmission lines were designed for all the RF input and output ports of the PCB. See figure 5.8 for schematic diagram of modulator board.

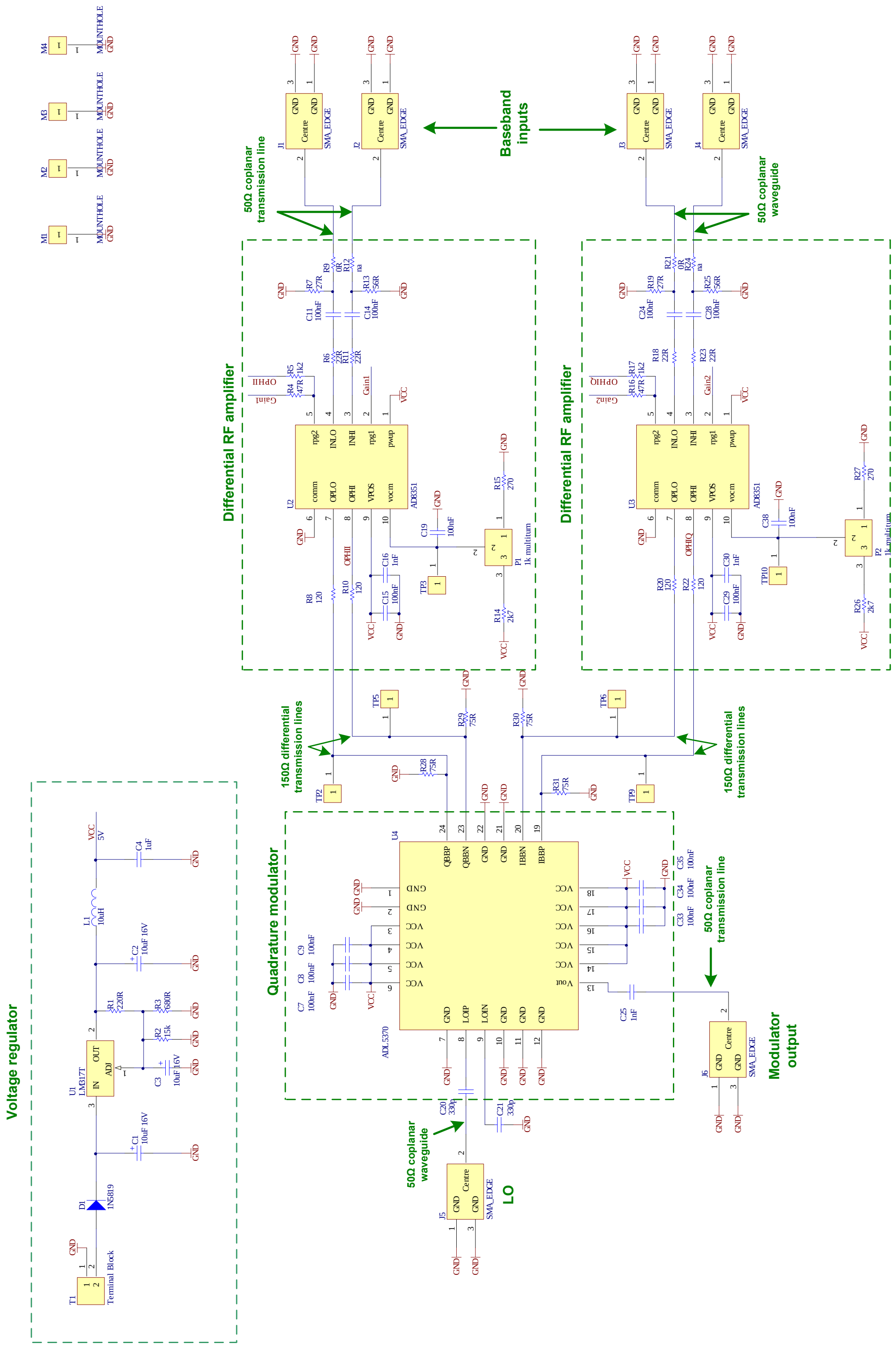

Voltage regulator
Differential RF amplifier
Differential RF amplifier
Quadrature modulator
Baseband inputs
50Ω coplanar transmission line
50Ω coplanar waveguide
150Ω differential transmission lines
150Ω differential transmission lines
50Ω coplanar transmission line
50Ω coplanar waveguide
Modulator output
LO
AD8351
AD8351
ADL5370
LM317T
SMA_EDGE
MOUNTHOLE

Figure 5.8: Passband PAM modulator schematic diagram

5.6 Synthesizer board

The synthesizer board is the local oscillator for the modulator. An integrated circuit from Analog Devices ADF4360-7 was used to generate the carrier signal at 720 MHz. It is an integrated integer-N synthesizer and VCO with a frequency range of 350 MHz to 1.8 GHz. A 10MHz crystal oscillator is used as a reference for the synthesizer.

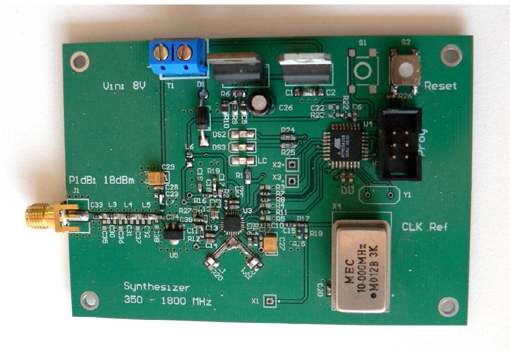


Figure 5.9: 720 MHz synthesizer board.

The free running frequency of the VCO is set with two inductors.

$$f_{vco} = \frac{1}{2\pi\sqrt{6.3 \times 10^{-12}(0.9 \times 10^{-9} + L_{ext})}} \tag{5.6.1}$$

Using equation 5.6.1, the center frequency of the VCO is set to 720 MHz with two 6.8 nH inductors. A phase frequency detector generates an error signal by comparing the reference generated clock with the internally divided VCO output. This error signal drives a charge pump. The output of the charge pump is filtered by an external loop filter that sets the loop dynamics of the synthesizer. The filtered charge pump signal is fed back to on-chip varactor diodes which pulls the VCO frequency in the direction that reduces the phase frequency error signal. A schematic diagram of the synthesizer board is shown in figure 5.10. The design of the loop filter was done using a CAD program ADIsimPLL from Analog Devices. The closed loop output frequency of the synthesizer is set by programming the prescaler and counters in the synthesizer. The output frequency is governed by the following equation:

$$f_{out} = [(P \cdot B) + A] \times f_{ref}/R \tag{5.6.2}$$

The R counter is a reference divider which sets the output frequency spacing. The $(P \cdot B) + A$ term in equation 5.6.2 sets the integer multiple with which the reference will be multiplied. The P parameter is a dual modulus prescaler and the A and B parameters represent two counters which sets divide ratios. The output frequency of 720 MHz is achieved with the following programming settings: P = 8; B = 90; A = 0; R = 10. The programming is done with a micro controller, the ATtiny26 from Atmel.

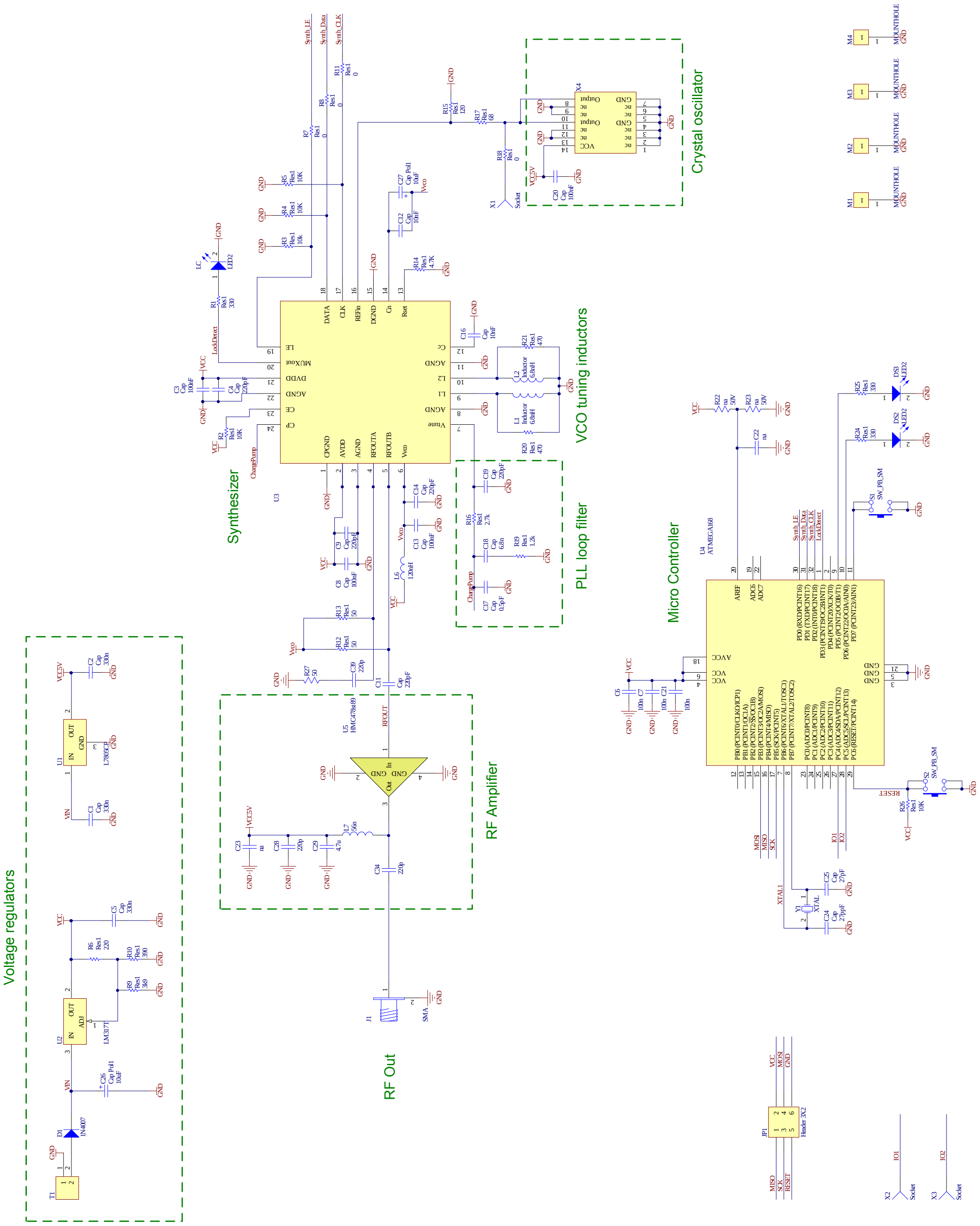


Figure 5.10: 720MHz synthesizer schematic diagram

Chapter 6

Passband PAM signal recovery

6.1 Overview of recovery processes

There are two main processes in the demodulator namely carrier phase recovery and the baseband timing recovery. Baseband synchronization was discussed in chapter 3 and will be implemented in the QPSK demodulator as discussed, except were otherwise stated. Continuing with the signal representation theory in section 5.2, the received passband PAM signal has the following representation:

$$\begin{aligned} r_{IF}(t) &= \cos(2\pi f_c(t-\tau))\sum_i a_i g(t-iT_s-\tau) \\ &\quad -\sin(2\pi f_c(t-\tau))\sum_i b_i g(t-iT_s-\tau)+w_{IF}(t) \end{aligned} \tag{6.1.1}$$

where τ is the channel delay and w_{IF} the channel noise. The next step is to decide how to get the information signal into the digital domain. It could be directly sampled at twice the IF frequency which is not practical for this application due to the high IF frequency and bandwidth of the data. Another possibility is to purposefully under-sample the IF signal and alias the modulated signal down to baseband. This topology requires a very stable clock at a very high frequency which is possible to generate but since synchronous sampling will be implemented, the clock frequency will vary and will thus also shift the aliased signal spectrum up and down. This option was thus not further explored.

A common topology typically implemented in high speed demodulators, is to convert the IF signal to a complex baseband signal. This is basically the reverse implementation of the modulator. Matched filters maximize the signal to noise ratio of the received signal and also remove the double frequency components. See figure 6.1.

Assume the frequency offset between the transmitter and receiver LO is defined as $v \triangleq f_c - f_{cL}$.

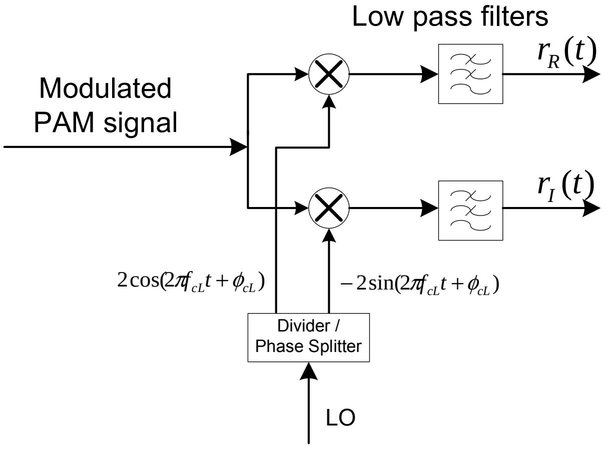


Figure 6.1: I/Q demodulator block diagram.

The output of the matched filters can expressed as a complex signal $r(t) \triangleq r_R(t) + jr_I(t)$.

$$r(t) = e^{j(2\pi vt+\theta)} \sum_i c_i h(t - iT_s - \tau) + n(t) \tag{6.1.2}$$

The noise term $n(t)$ is composed of the real and imaginary low pass filter noise outputs $n(t) = n_R(t) + jn_I(t)$. The phase shift θ is caused by the channel delay and the phase difference between the transmitter oscillator and the local oscillator phase. $\theta = \theta_{cL} - 2\pi v\tau$. The symbol c_i is the complex baseband data.

From equation 6.1.2 there are four unknown parameters (v, θ, T_s, τ) in addition to the data symbols. For successful demodulation, these four parameters need to be estimated. Estimating T_s and τ have already been discussed which leaves v and θ. Sampling this complex baseband signal at $t = kT_c + \hat{\tau}$ produces equation 6.1.3.

$$r(kT_c + \hat{\tau}) = e^{j(2\pi v(kT_c+\hat{\tau})+\theta)} \sum_i c_i h(kT_c + \hat{\tau} - iT_s - \tau) + n(kT_c + \hat{\tau}) \tag{6.1.3}$$

The matched filters satisfy Nyquist's first criterion which means that there is no ISI caused by the pulse shaping filters. Equation 6.1.3 can thus be simplified to

$$r(kT_c + \hat{\tau}) = e^{j(2\pi v(kT_c+\hat{\tau})+\theta)} c_i h(kT_c + \hat{\tau} - iT_s - \tau) + n(kT_c + \hat{\tau}) \tag{6.1.4}$$

Assume in the discrete time domain that $r(k) = r(kT_c + \hat{\tau})$. Also let the pulse shape amplitude $h(kT_c) = c_{amp}$ and assume there is perfect baseband timing recovery.

$$r(k) = c_{amp} c_k e^{j(2\pi vkT_c+\theta_{cL})} + n(k) \tag{6.1.5}$$

The constant c_{amp} refers to the magnitude of the baseband symbol. An AGC circuit was designed to normalize c_{amp}. It is discussed in section 6.3.3. The purpose of the carrier phase recovery algorithm is to remove the complex exponential factor in equation 6.1.5. The complex exponential introduces an unwanted rotation of the constellation diagram. It should be clear that if $v = 0$ and $\theta_{cL} = 0$ only the baseband symbol with some noise remains. For this condition to be realized, the local oscillator frequency and phase must be equal to the received modulated signal's frequency and phase.

Carrier recovery is implemented with a dual locking system. Digital carrier recovery is implemented inside the FPGA for initial locking. When the internal carrier lock detector detects a locked condition, the integrators in the internal recovery loop is ramped down at a slow enough rate that the external carrier recovery loop can track the frequency and phase offset. This carrier recovery scheme has the advantage that a wide loop bandwidth can be used for initial locking because there is very little loop delays in the internal carrier recovery loop. The external recovery loop has a low loop bandwidth for phase tracking. The external recovery loop programs a DDS which generates a 200 MHz carrier. The LO for coherent down conversion is generated by mixing the DDS output with a 1240 MHz from a synthesizer. The 1440 MHz mixing product is filtered and sent to an I/Q demodulator which internally divided it by two to produce the 720MHz carrier. The carrier recovery loop thus has the configuration shown in figure 6.2.

The design process of the carrier recovery loop is very similar to the timing recovery loop. The carrier recovery loop can also be modeled as a linear discrete time phase lock loop. The quadrature down converter together with the phase error detector forms the classic phase detector in a phase lock loop. The carrier loop filter sets the loop dynamics such as bandwidth and overshoot and the DDS performs the function of the VCO in a PLL.

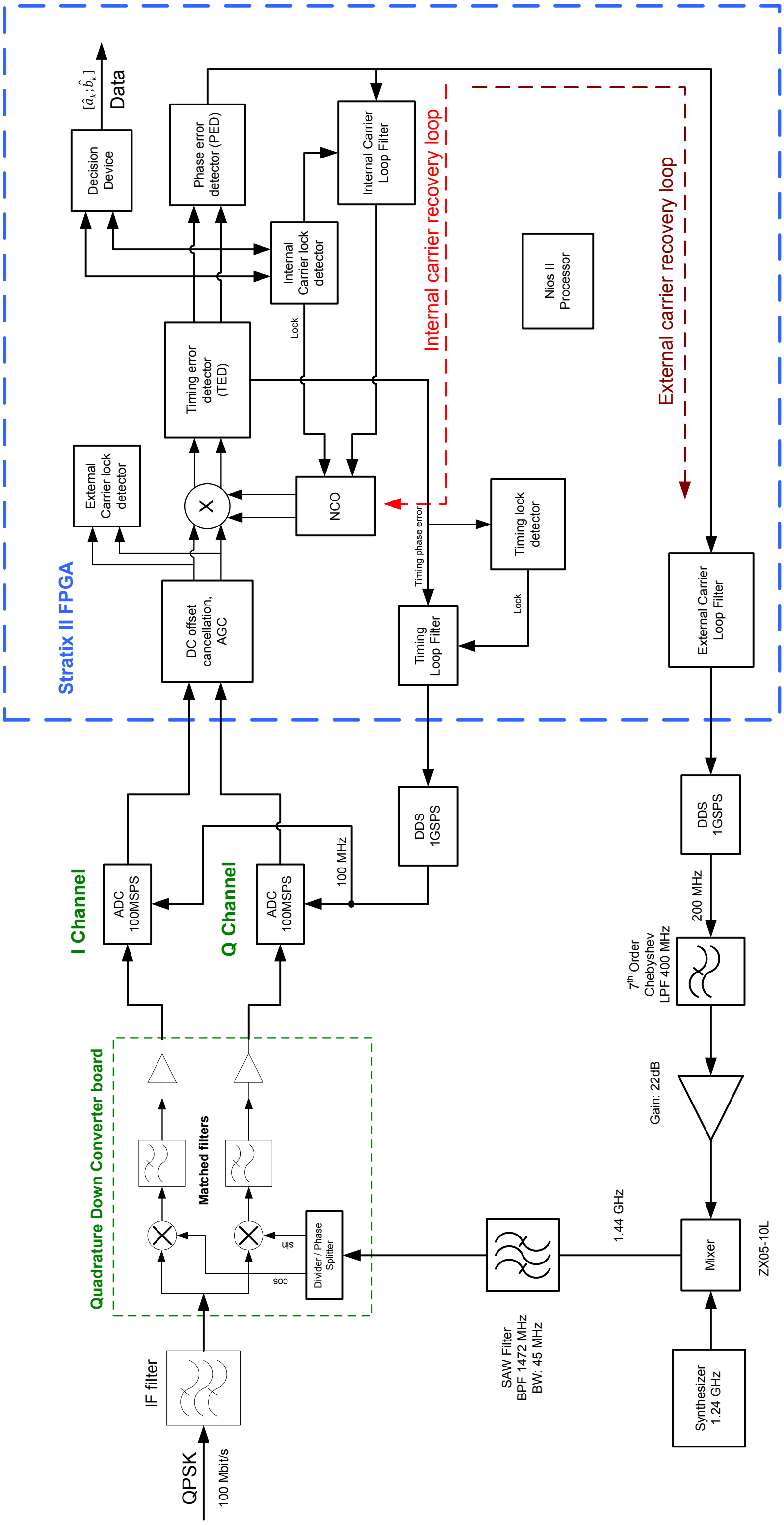

Stratix II FPGA
Data
$[\hat{a}_k;\hat{b}_k]$
Decision Device
Phase error detector (PED)
Internal Carrier Loop Filter
Internal Carrier lock detector
Lock
Timing error detector (TED)
External Carrier lock detector
NCO
Internal carrier recovery loop
External carrier recovery loop
Nios II Processor
Timing phase error
Timing lock detector
Lock
DC offset cancellation, AGC
Timing Loop Filter
External Carrier Loop Filter
I Channel
ADC 100MSPS
Q Channel
ADC 100MSPS
100 MHz
DDS 1GSPS
DDS 1GSPS
200 MHz
7th Order Chebyshev LPF 400 MHz
Gain: 22dB
Quadrature Down Converter board
Matched filters
sin
cos
Divider / Phase Splitter
IF filter
QPSK
100 Mbit/s
SAW Filter BPF 1472 MHz BW: 45 MHz
1.44 GHz
Mixer
ZX05-10L
Synthesizer 1.24 GHz

Figure 6.2: Demodulator RF block diagram

6.2 Internal carrier recovery

The internal carrier recovery is implemented purely inside the FPGA. The major advantage of this implementation is that the loop bandwidth can be very wide as there are not any extra blocks in the feedback loop with large latencies that limit the bandwidth. The multiplier block in figure 6.3 is a complex multiplier which shifts the frequency spectrum of the received QPSK signal. The timing error detector extracts the peak samples from the I and Q channels. The phase error detector generates an error signal which steers the loop in the direction that will reduce the carrier phase error. The error signal is filtered by the internal loop filter which controls the numerically controlled oscillator (NCO). When the carrier lock detector detects a locked condition, the integrators in the loop filter and the NCO are slowly ramped down to zero.

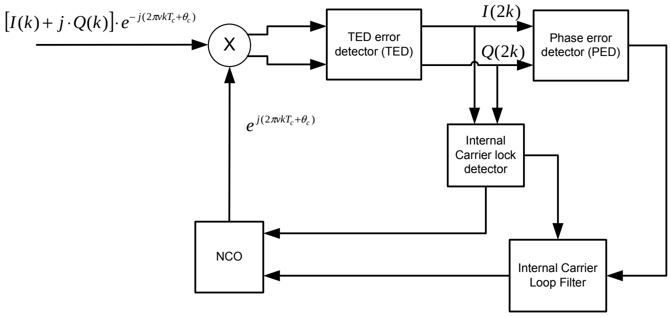

$[I(k)+j\cdot Q(k)]\cdot e^{-j(2\pi vkT_c+\theta_c)}$
X
TED error detector (TED)
$I(2k)$
$Q(2k)$
Phase error detector (PED)
$e^{j(2\pi vkT_c+\theta_c)}$
Internal Carrier lock detector
NCO
Internal Carrier Loop Filter

Figure 6.3: Internal carrier recovery.

6.2.1 Phase error detector

An important task in the development of the QPSK demodulator is the development of the carrier recovery algorithm. A phase lock loop can be viewed as a maximum likelihood phase estimator [4]. A maximum likelihood phase estimator for QPSK is derived in addendum B. The process involved writing a likelihood function $f_{\mathbf{z}/\theta,H_i}(z_1, z_2/\theta, H_i)$ for the error vector of a QPSK signal given a phase error θ. The next step was averaging this likelihood function over the four possible symbols to make the function independent of the transmitted symbol. The result is the likelihood function $f_{\mathbf{z}/\theta}(z_1, z_2/\theta)$. To determine the maximum of the likelihood function, the partial derivative of the log-likelihood function is calculated with respect to θ and equated to zero. When this partial derivative is zero, $\theta = \hat{\theta}_{ML}$. The result is shown in equation 6.2.1.

$$\begin{aligned}&\tanh\left(C_2\int_0^{T_s} y(t)\cdot\sin(\omega_c t+\theta)dt\right)\cdot C_2\cdot\int_0^{T_s} y(t)\cdot\cos(\omega_c t+\theta)dt\\ -&\tanh\left(C_2\int_0^{T_s} y(t)\cdot\cos(\omega_c t+\theta)dt\right)\cdot C_2\cdot\int_0^{T_s} y(t)\cdot\sin(\omega_c t+\theta)dt/_{\theta=\hat{\theta}_{ML}}=0 \qquad (6.2.1)\end{aligned}$$

The constant C_2 has the value $\frac{2A}{N_0}$. This constant complicates the design somewhat because the carrier recovery loop needs to know the signal to noise ratio or just the noise power since the AGC will normalize the input signal. Equation 6.2.1 can now be drawn in block diagram form to implement carrier recovery as shown in figure 6.4. There is a loop filter which is not defined

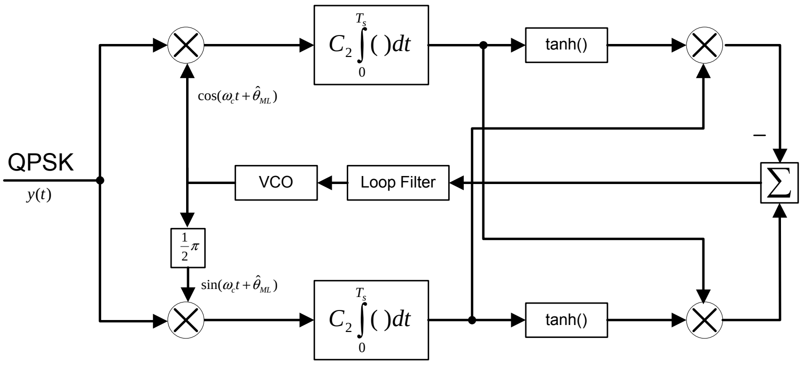


Figure 6.4: Carrier recovery control loop

in equation 6.2.1. This filter is needed because the control loop operates with an iterative process to make the phase estimate. It contains an integrator to accumulate the the phase error and helps to set the control loop behavior such as bandwidth and overshoot. Equation 6.2.1 contains two tanh() functions which is either very resource intensive or slow to calculate. The simplest implementation for the *tanh*() is to replace it with a signum function. The Costas loop is a carrier recovery method which uses the signum implementation. A better approximation is a saturation block with the characteristics shown in equation 6.2.2.

$$Saturation(x) = \begin{cases} x & for \quad |x| < 1 \\ sign(x) & for \quad |x| \geq 1 \end{cases} \tag{6.2.2}$$

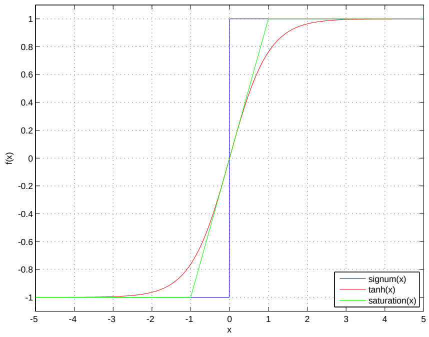


Figure 6.5: Tanh function approximations.

Keeping in mind that all the blocks described in this thesis had to be programmed into an FPGA, it was decided to implement the tried and tested Costas loop with the signum function. The signum implementation does not require the scaling with constant C2. Another problem that is avoided is having to recalculate the loop filter coefficients each time the signal to noise ratio changes because the loop gain changes when C2 change. The other implementations are however simulated to evaluate their performance.

A QPSK signal has four $\pi/2$ phase ambiguities. The purpose of the PED is to create a error signal that has stable lock points with the following characteristics.

1. Stable lock points produce zero output at multiples of $\pi/2$.

2. The error signal must have a positive slope through the stable lock points.

The outputs of the matched filters are multiplied by C2 which results in baseband signals with amplitudes of $\frac{2 \cdot E_s}{N_0}$ which is a function of the signal to noise ratio. All three implementations have the same four locking points. The lock points are on the x-axis and the S-curves have positive slopes at the lock points. Three simulations showing the S-curves of the different PED implementations at three different noise levels are shown in figure 6.6. It is evident that the three implementations converge as the signal to noise ratio is increased.

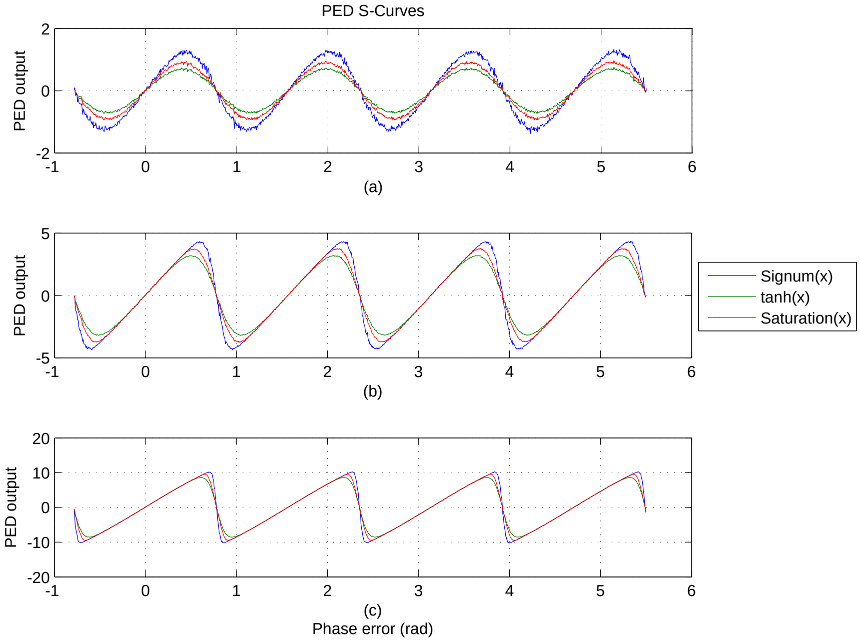


Figure 6.6: Phase error detector S-curve simulations with (a) $\frac{E_s}{N_0} = 3dB$, (b) $\frac{E_s}{N_0} = 6dB$ and (c) $\frac{E_s}{N_0} = 9dB$.

As with the timing error detector, the gain of the phase error detector is determined by the slope of the S-curve. The derivative of the phase error detector output is shown in figure 6.7. The PED gain is at its maximum at the locking points with a value of two.

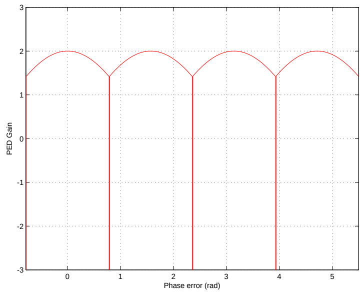


Figure 6.7: Phase error detector gain.

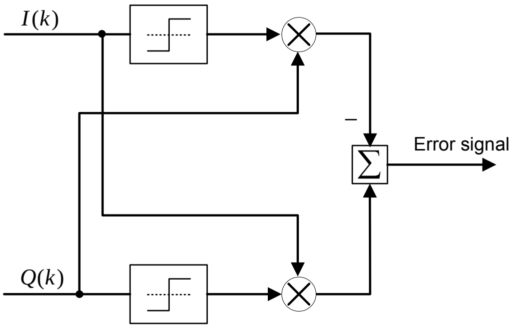


Figure 6.8: Phase error detector.

6.2.2 Internal carrier recovery loop filter

The design of the loop filter coefficients for the carrier recovery loop follows the same procedure used for the timing recovery loop in section 3.6. The structure of this first order loop filter basically consists of two multipliers and an integrator with some additional logic to empty the integrator when triggered. Constants C1 and C2 sets the filter response and C3 determines the ramp rate at which the integrator is emptied. When the internal recovery loop is not in lock, switch 1 is closed and switch 2 is open until the carrier lock detector detects the locked condition and both switches are toggled. The integrator contains the accumulated frequency offset of the received QPSK signal. When the integrator is ramped down, the frequency offset is transferred to the external loop filter which programs the DDS. The calculation of the feedback gain is shown in the design procedure of the external carrier recovery loop and influences the phase lag of the external carrier loop when ramping down.

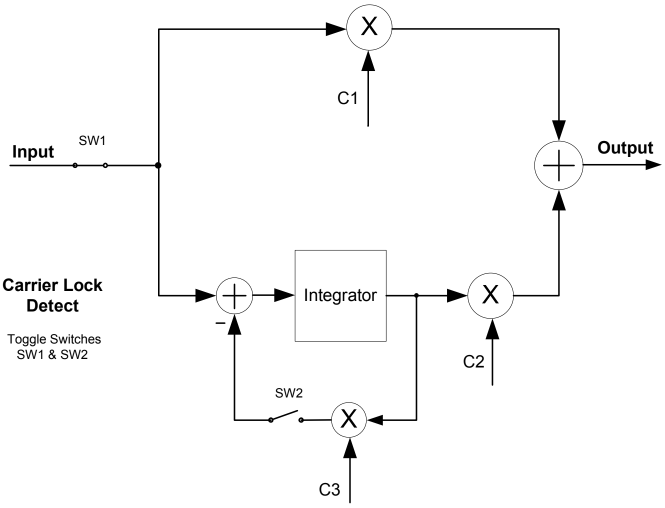


Figure 6.9: Internal carrier loop filter

6.2.3 Numerically controlled oscillator

The numerically controlled oscillator generates the in-phase sine and cosine carriers to frequency shift the baseband I and Q data to zero hertz. The frequency of the NCO is controlled by the internal carrier loop filter. When the internal carrier recovery loop is locked, any offset frequency (v) is slowly ramped down. When the frequency offset is zero, there is still a residual phase offset in the integrator of the NCO.

$$NCO_{out}(k) = \exp\left(j2\pi vk + \phi\right) = \cos(2\pi vk + \phi) + j \cdot \sin(2\pi vk + \phi) \qquad (6.2.3)$$

Consider the case where the internal carrier recovery loop is locked and the internal loop filters integrator has been ramped down. There will be a rotation of the baseband I and Q signals at the ADC's. The external carrier recovery loop will not be able to remove this phase offset at the ADC's input because the rotation is removed between the ADC's and the phase error detector. The solution is to also ramp the phase offset to zero when the lock detectors output is high. The sine output is used as feedback because it is zero when the phase offset is zero. The sine output is bounded in the range of -1 to +1 which means that when the feedback gain is set very low, the integrator will not be able to integrate out the phase offset. An additional integrator is inserted to accumulate any phase offset but the extra integrator causes the feedback loop to oscillate. Some proportional feedback (*C5*) is inserted to stabilize the feedback loop.

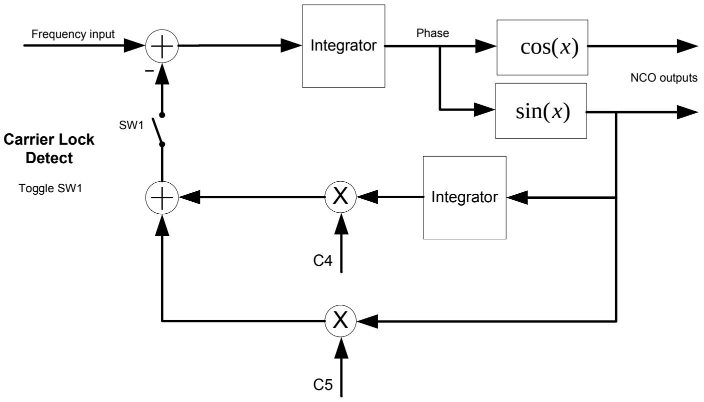


Figure 6.10: Numerically controlled oscillator

6.2.4 Internal carrier lock detector

The carrier lock detector determines when the phase error of the I and Q signals is below a certain threshold. There are various carrier lock detection algorithms used. In [7] several algorithms are evaluated. The Y_5 algorithm is selected, because it has the highest probability of detection (P_d) at low signal to noise ratios. The lock observation is generated with equation 6.2.4.

$$\begin{aligned} z &= \sum_{k=1}^{N} Y_5(k) && (6.2.4) \\ &= \sum_{k=1}^{N} 2\left|I(k)\cdot Q(k)\right| - \left|I(k)^2 - Q(k)^2\right| && (6.2.5) \end{aligned}$$

The parameter N in 6.2.4 sets the observation interval. The longer the observation, the lower the signal to noise ratio of the QPSK signal maybe for a given P_d. In the unlocked state, the mean $\mu_0 = 0$ and the variance σ_0 is calculated with equation 6.2.6 with constant A equal to the QPSK baseband signal amplitude and σ the noise for each channel. There is no closed form solution for the mean and variance of $Y_5(k)$ in the locked condition and is determined through simulation.

$$\sigma_0 = \left(16 - \frac{32}{\pi}\right)\cdot\left(2\sigma^4 + 4\sigma^2 A^2 + A^4\right) \qquad (6.2.6)$$

When $z > \lambda_{th}$, then the lock detector output a high signal indicating the locked condition. Figure 6.11 shows a practical implementation of the lock detector.

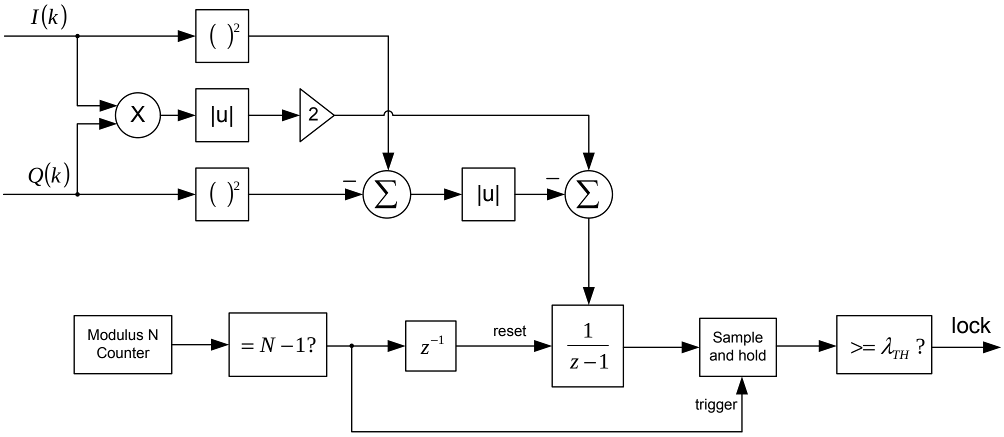


Figure 6.11: Internal Carrier Lock Detector

6.3 External carrier recovery

The external carrier recovery loop is used to implement coherent frequency down conversion in the demodulator. At first glance, the external carrier recovery loop would seem a little redundant as the internal carrier recovery loop digitally removes any frequency offset from the baseband I and Q signals. The problem is that when there is a frequency offset in the complex baseband data, the frequency response of the matched filters are miss aligned with the frequency spectrum of the received baseband signals. The frequency offset causes the peaks of the matched filter outputs to be reduced and so decreases the signal to noise ratio of the baseband signals. The distortion is similar to inter-symbol interference and increases the bit error rate of the demodulator. Figure 6.12 shows the eye diagram of the I channel matched filter output without a frequency offset and Figure 6.13 shows the case where the baseband data is given a 1 MHz offset.

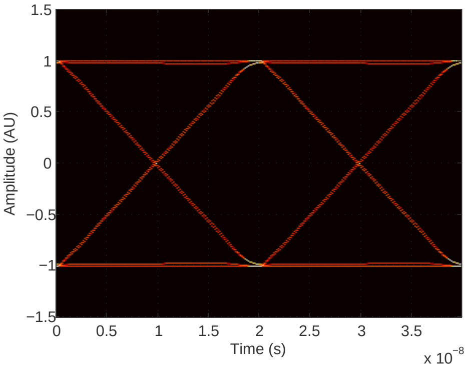


Figure 6.12: Eye diagram with zero baseband frequency offset.

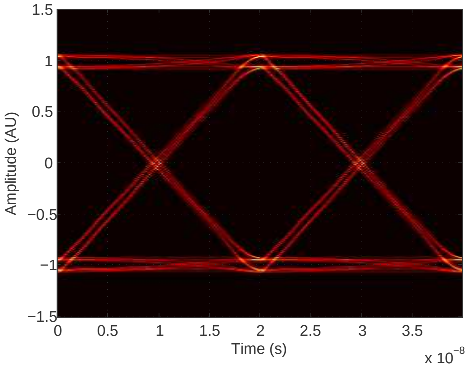


Figure 6.13: Eye diagram with 1 MHz baseband frequency offset.

It is clear that the eye opening in the eye diagram is reduced when there is a frequency offset. To avoid this degradation of the baseband data, any frequency offset must be removed before the baseband data is filtered by the matched filters. The external carrier recovery loop is shown in figure 6.14.

The first block in the recovery chain is the I/Q down converter which frequency translates the QPSK signal to baseband. Two matched filters maximize the signal to noise ratio of the baseband signals. The two ADC's sample the I and Q baseband signals and send the digital data to the FPGA. The ADC's are the FPGA's on board ADC's and are discussed in section 3.3. In the FPGA, any DC offset is removed from the I and Q channels and also normalizes the baseband signals with the digital AGC block. The symbol timing block decimates the baseband data so that only the peak samples of the matched filters are passed on to the complex multiplier of the internal carrier recovery block. The external carrier recovery lock detector determines when the phase error of the I and Q channels are below a maximum threshold and is a duplicate of the lock detector used in the internal carrier recovery loop. The phase error detector has already been discussed in section 6.2.1. The phase error signal is filtered by the external loop filter and then

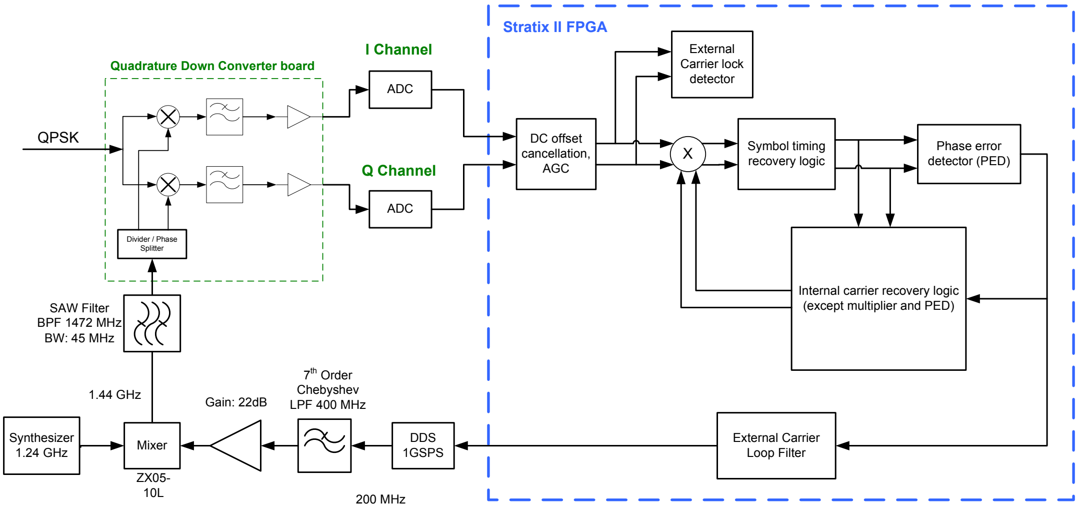

Stratix II FPGA
I Channel
Quadrature Down Converter board
QPSK
ADC
Q Channel
ADC
Divider / Phase Splitter
External Carrier lock detector
DC offset cancellation, AGC
X
Symbol timing recovery logic
Phase error detector (PED)
Internal carrier recovery logic (except multiplier and PED)
SAW Filter BPF 1472 MHz BW: 45 MHz
1.44 GHz
Gain: 22dB
7th Order Chebyshev LPF 400 MHz
Synthesizer 1.24 GHz
Mixer
ZX05-10L
DDS 1GSPS
200 MHz
External Carrier Loop Filter

Figure 6.14: External carrier recovery loop

programmed into the DDS through the parallel programming interface. The DDS and following filter is the same PCB which is used in the symbol recovery loop except for the clock distribution section which are not populated. The DDS board produces a 200 MHz carrier which is amplified to generate a large IF input signal for the mixer. A synthesizer generates the 1.24 GHz LO signal for the mixer which together with the IF signal generate the 1.44 GHz RF output. A SAW filter is the final block in the feedback loop and removes all the mixing products except for the 1.44 GHz component from the mixer output.

6.3.1 I/Q demodulator board

The I/Q demodulator also known as a quadrature down converter performs the down conversion of the IF QPSK signal to complex baseband. After the QPSK signal is down converted matched filters maximizes the signal to noise ratio before the baseband signals are amplified by differential amplifiers. This board is basically the reverse implementation of the modulator board discussed in section 5.5. The same differential amplifiers as the modulator board are used to amplify the I and Q channels. The matched filters are built on separate modular boards which are then in turn soldered onto the down converter board. The filters are implemented as discussed in section 3.2. See figure 6.16 for schematic diagram of down converter board.

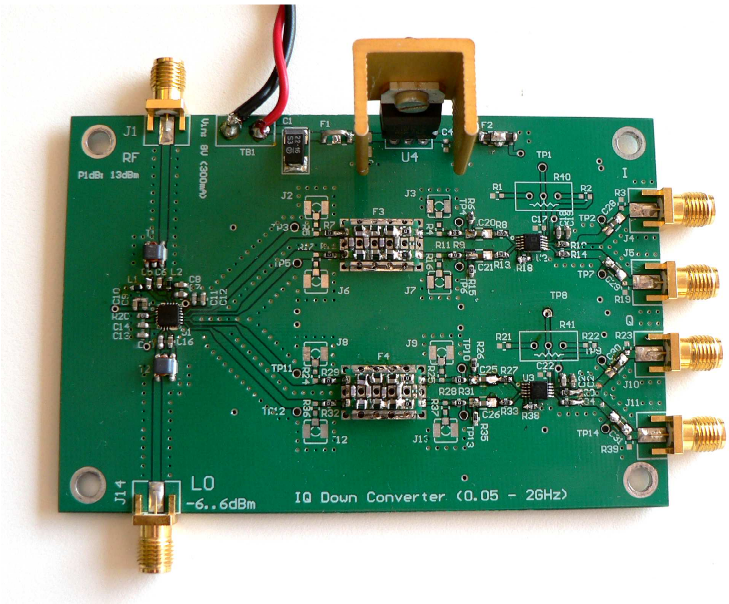


Figure 6.15: I/Q demodulator board.

The integrated circuit used to perform the quadrature down conversion is the AD8471 from Analog devices. It has a baseband bandwidth of 600 MHz which is more than enough and a RF operating range from 0.1 - 4 GHz. The LO port has an internal frequency divider and phase shifter to generate two orthogonal 720 MHz from the 1.44 GHz carier. The RF and LO ports on the down converter IC are differential ports. Baluns from ETC (1-1-13) are used to convert the single ended LO and RF signals to differential ones. The RF port has two 120nH inductors for input matching. High quality 0603 EPCOS inductors are used in the matching network as lower quality inductors create a mismatch at the input which severely degrades the eye diagram opening of the baseband outputs.

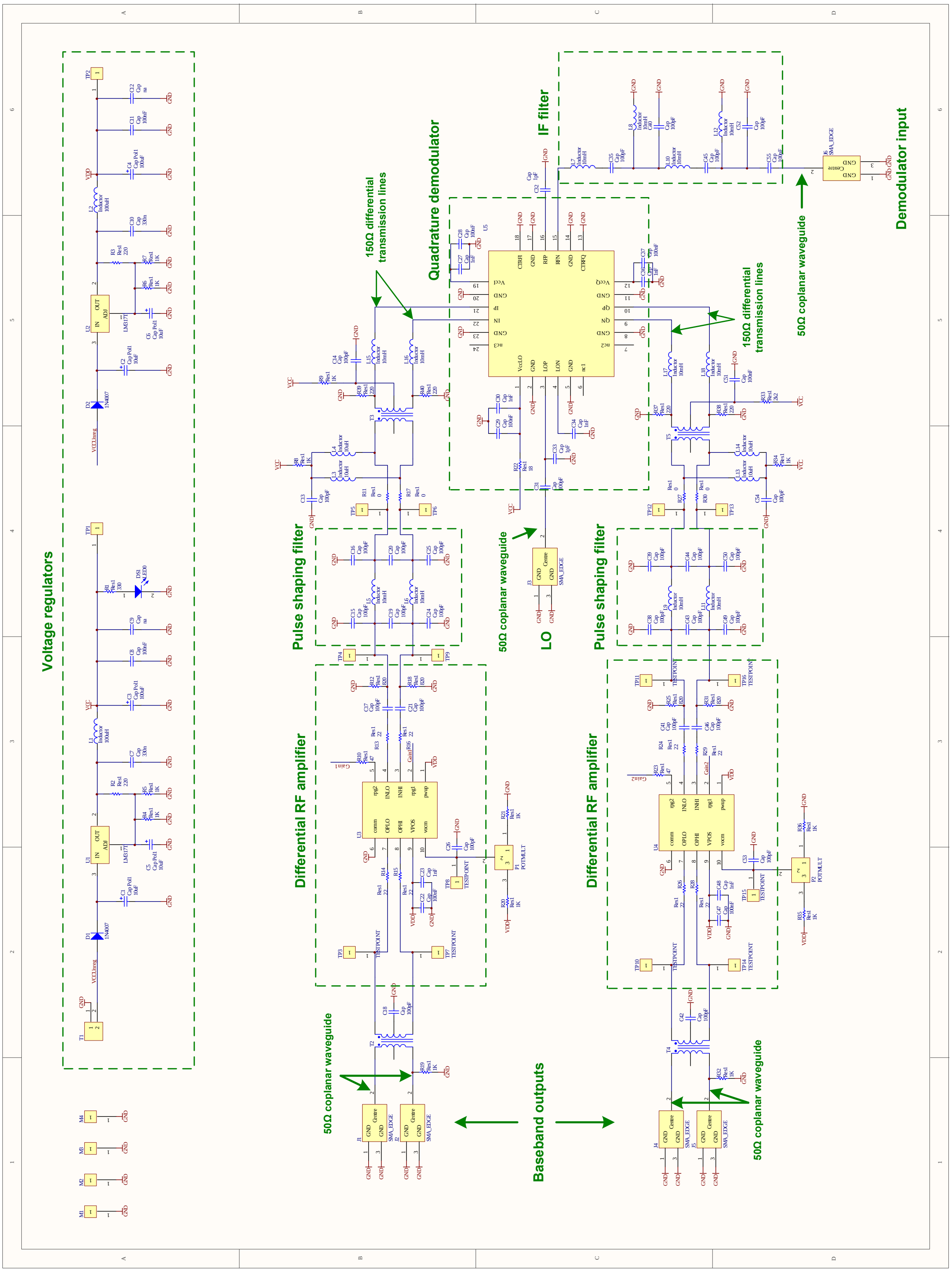


Figure 6.16: Passband PAM demodulator schematic

6.3.2 DC Compensation

The sampled matched filter I and Q signals typically have small DC offsets caused by carrier leak-through at the transmitter or from a slight voltage offset in the analog to digital converter differential inputs. A DC offset in either the I or Q channel will cause the AGC block to overestimate the power of the received QPSK signal and thus the AGC gain will be too low. Another obvious problem the DC offset causes is the detection bias in the symbol detector. The optimum threshold for the symbol detector is the mean of the individual I and Q signals which is assumed to be zero. If there is a DC offset in the I or Q signals, the mean is not zero anymore. This will introduce a bias for one of the two symbols which results in $P(e|S1) \neq P(e|S2)$ and a higher total $P(e)$ than when there is no offset.

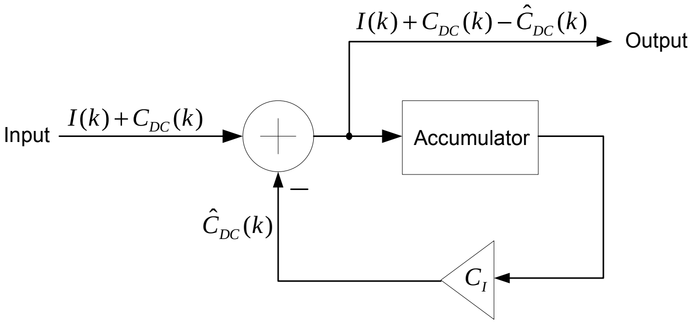


Figure 6.17: DC Compensator

The DC compensator block is a maximum likelihood estimator which converges to the mean of the input and subtracts it from the input signal. It can also be viewed as a first order high pass filter. The Bode plot for the filter is shown in figure 6.18. The gain (C_I) is set to 2^{-18} which means

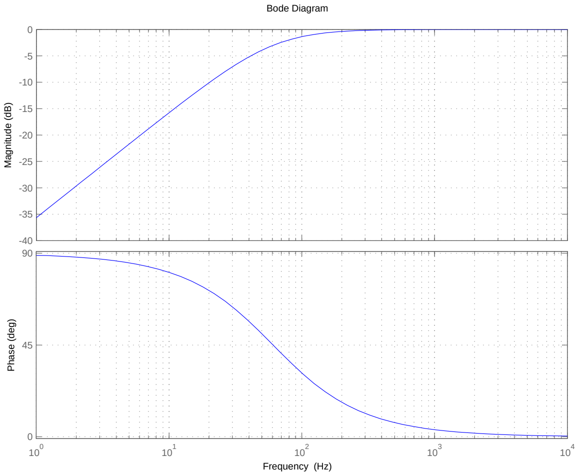


Figure 6.18: DC filter bode plot

that the lower 18 bits of the accumulator are simply thrown away. The -3dB cut-off frequency of the high pass filter is 61 Hz.

$$f_{-3dB} = \frac{C_I}{2\pi T_s} \tag{6.3.1}$$

6.3.3 Automatic gain control

The purpose of the automatic gain control block is to normalize the amplitude of the baseband symbols. Normalizing the symbol amplitudes in turn keeps the loop gains of the carrier and phase recovery blocks constant irrespective of what the analog received signal amplitude is. An AGC circuit could be implemented in hardware, software or a combination of the two.

The AGC implemented in this demodulator is a digital AGC. This implementation has the drawback that when the input signal is small, the effective number of bits used by the analog to digital converter is less than what it would have been if analog amplification was used to correct the amplitude of the baseband signal. The result is an increase in quantization noise. Thus there is a limit where digital scaling of the symbols also scales the quantization noise to the extent that the bit error rate of the demodulator exceeds the system specification. The advantage of the digital AGC is that it is easier to implement since no hardware is needed. Figure 6.19 shows the architecture of the AGC block. Calculating the power of the complex baseband input is not possible since the inputs are sampled below the Nyquist sampling rate. When the symbol timing recovery loop is locked, the zero crossing samples contribute negligible power compared to the peak samples. Thus only the peak samples are used in the power estimation. The peak samples are extracted after the symbol timing recovery block and that is where the feedback is taken. The square of the I and Q peaks are added together and subtracted from the reference power level. The power error is accumulated and scaled before multiplying the inputs with the ratio of the reference amplitude and the baseband signal amplitude. Already one would start to wonder

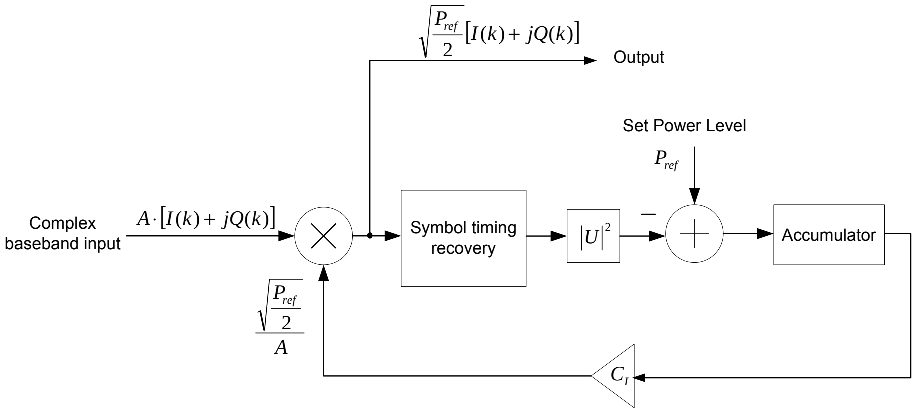


Figure 6.19: AGC block

what happens when the symbol timing loop is not locked. The peak sample estimate slides over the whole symbol period. First consider the case where the data is alternating. The average of the symbol pulse shape amplitude would be used in the power estimation.

$$\begin{aligned} P_{unlock} &= I_{avg}^2 + Q_{avg}^2 \\ &= 0.25 \cdot I_{peak}^2 + 0.25 \cdot Q_{peak}^2 \end{aligned} \tag{6.3.2}$$

Now consider the case when the data is random. The probability of a transition is 50%. If there is no transition, the amplitude is at the peak value. The average estimated power is then shown in 6.3.3. The AGC block will underestimate the power of the complex baseband signal by 1.6 times which means the AGC will have 1.6 times the gain that it should have.

$$\begin{aligned} P_{unlock} &= 0.5 \cdot I_{avg}^2 + 0.5 \cdot I_{peak}^2 + 0.5 \cdot Q_{avg}^2 + 0.5 \cdot Q_{peak}^2 \\ &= 0.625 \cdot I_{peak}^2 + 0.625 \cdot Q_{peak}^2 \end{aligned} \tag{6.3.3}$$

One possible solution is to scale the reference power by 0.625 when the symbol recovery loop is not locked. When the symbol recovery loop is locked, the scaling factor is removed. This type of cascaded locking is a bit elaborate. The carrier and symbol recovery loops are second order systems with closed loop poles on the left half of the S-plane or within the unit circle in the Z-domain. This means that the feedback loop will not become unstable with a bit of extra loop gain. The most noticeable effect with the extra gain is the loop bandwidth of the carrier and symbol recovery loops will be increased and will correct itself when the symbol recovery loop is locked. The final implementation of the AGC simply allows the AGC to lock while concurrently locking the symbol and carrier recovery loops.

6.3.4 External carrier lock detector

The external carrier lock detector is a duplicate of the detector used in the internal carrier recovery. Refer to section 6.2.4 for the design process.

6.4 Passband signal recovery simulation overview

The passband signal recovery simulation operates on the same principle as the baseband simulations where the clock of the demodulator is held constant and the transmitter′s data clock is varied to lock the transmitter to the receiver. The simulation model consists of a transmitter which generates a complex baseband signal and a QPSK demodulator with carrier and symbol recovery blocks. The symbol recovery is basically the same as the simulation setup in section 4.1 except there for the PED′s that are duplicated for each baseband channel and summed. The Matlab Simulink model for the QPSK demodulator is shown in figure 6.21. Figure 6.20 shows a simulation where the carrier and symbol recovery loops simultaneously lock.

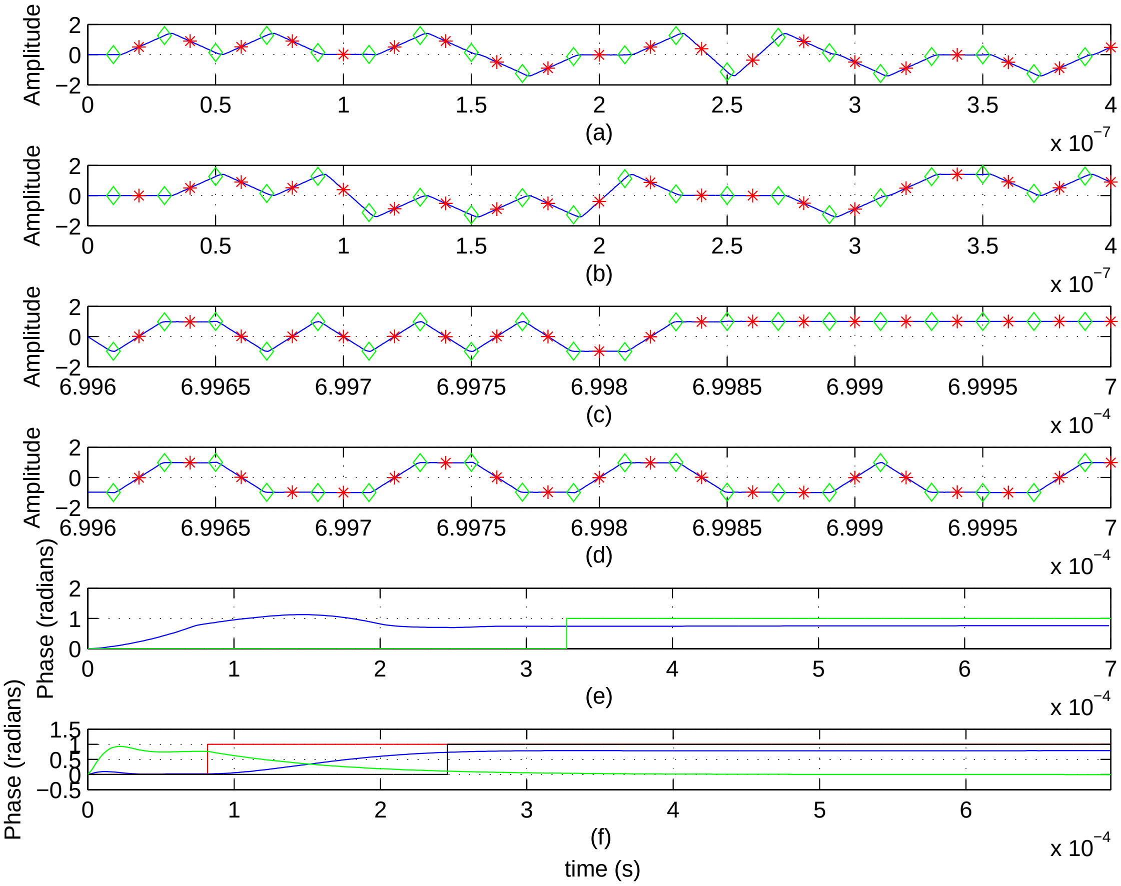


Figure 6.20: QPSK demodulator simultaneous carrier and symbol recovery simulation.

Subplots (a) and (b) of figure 6.20 are the initial I and Q baseband signals at the input of the ADC′s. The red stars indicate the estimated zero crossings and the green diamonds the peaks. Clear the demodulator is not locked as the estimated samples are not where they should be.

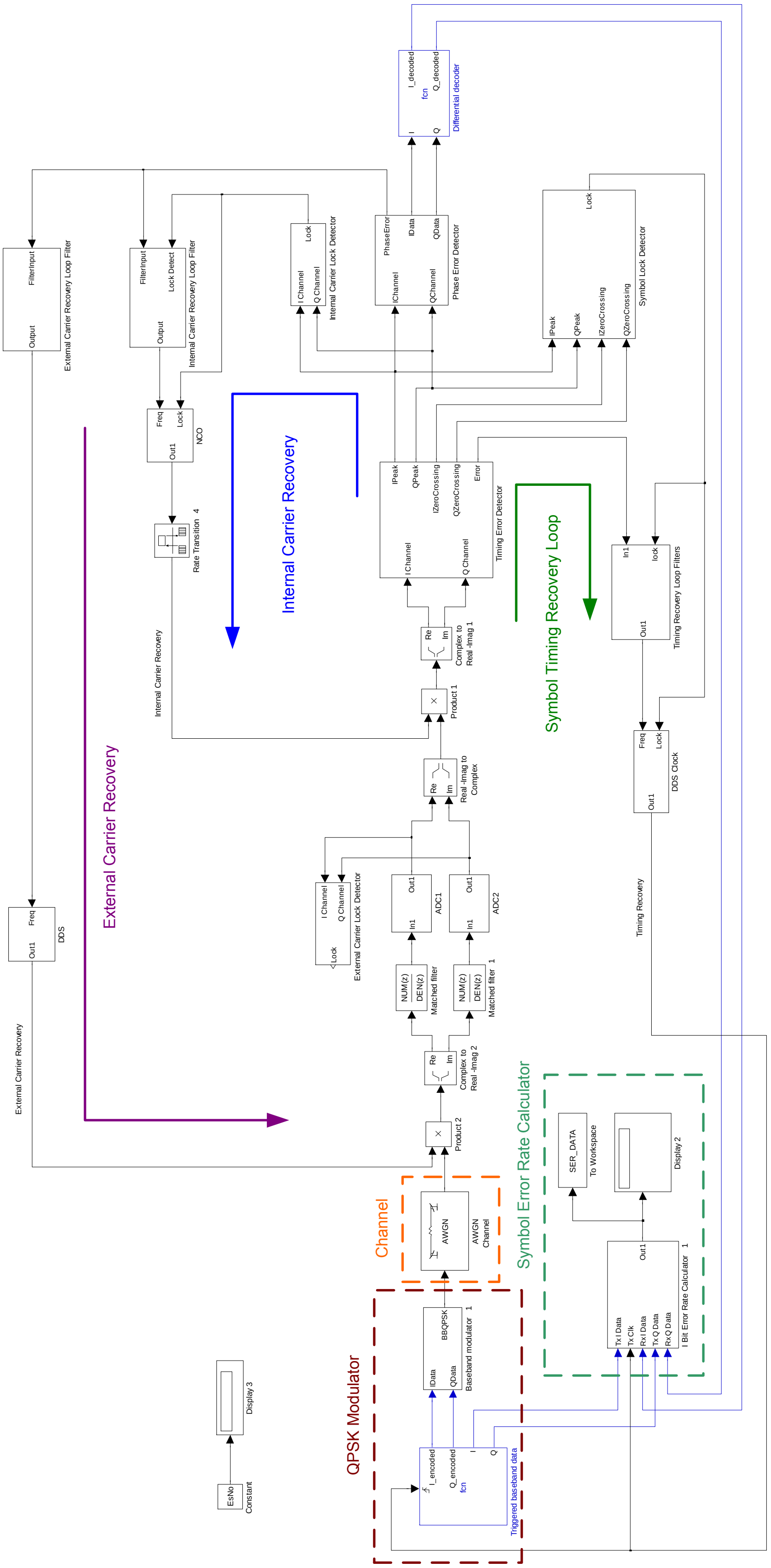


Figure 6.21: QPSK demodulator Simulink model.

Also, the I and Q channels are rotated by a phase offset which causes the baseband signals to have more than two peak amplitudes for the triangles. In subplots (c) and (d) the demodulator is locked and the estimated peaks and zero crossings are on the peaks and zero crossings of the I and Q baseband signals. The blue trace in subplot (e) is the phase step response of the symbol recovery DDS. The green trace is the symbol lock detector output. Subplot (f) is carrier recovery phase response for the internal and external recovery loops. The green trace is the phase of the numerically controlled oscillator (NCO) of the internal carrier recovery loop. It internal carrier recovery loop has a 100 kHz bandwidth and locks quickly. The red trace is the output of the internal carrier lock detector. When it detects a locked condition, the frequency and phase of the NCO is ramped slowly to zero. As the phase ramps down the external carrier recovery loop compensates for the phase offset. The blue trace shows how the phase of the external carrier recovery DDS increases as the NCO's phase decreases.

There are three phase error detectors discussed in section 6.2.1. The step responses of the internal carrier recovery loop with the three PED's are shown in figure 6.22.
There is very little difference in the shape of the step responses of the three PED's and all are within 1.5 kHz of the designed 100 kHz loop bandwidth. The external carrier recovery loop however has about the same loop delay as the symbol recovery loop which limit the maximum bandwidth of the recovery loop. An external loop bandwidth of 10 kHz is simulated in figure 6.23.

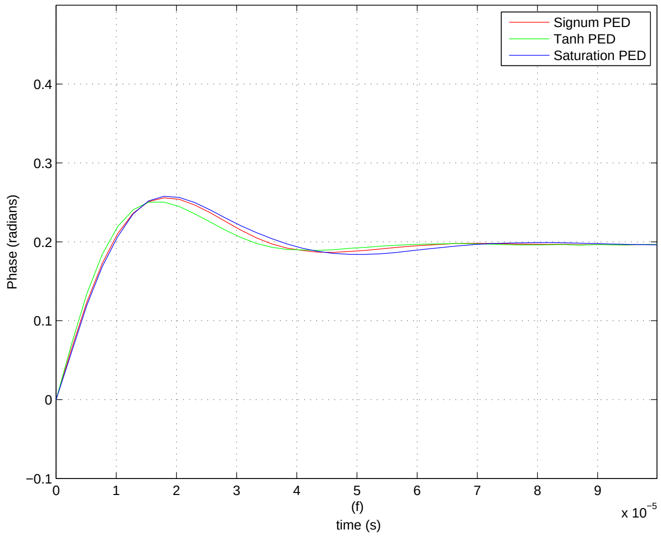


Figure 6.22: Internal carrier recovery phase step responses of different PED's.

The carrier symbol synchronization process can be viewed by looking at the scatter diagram of the I and Q channels. Figure 6.24. As the carrier recovery loop locks the scatter diagram is rotated to the lock position.

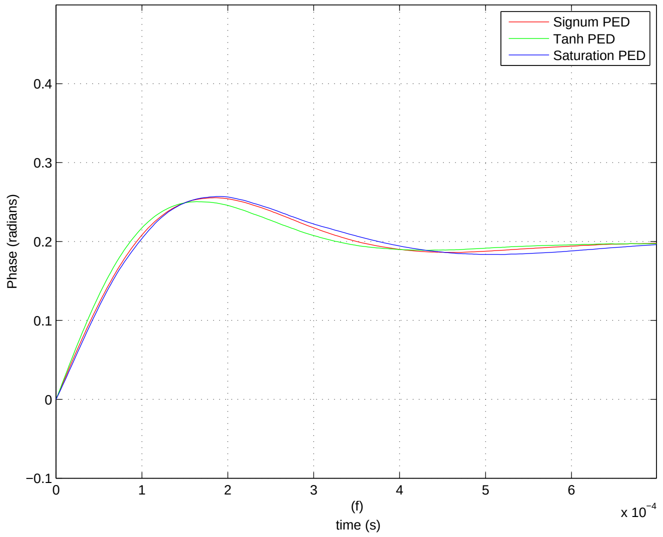


Figure 6.23: External carrier recovery phase step responses of different PED's.

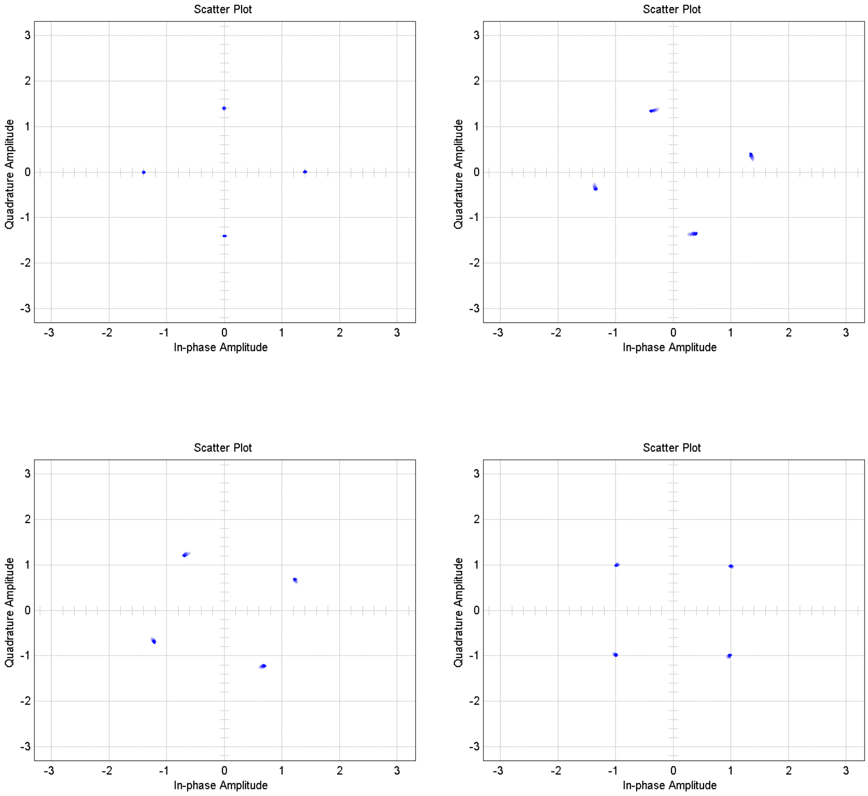


Figure 6.24: Scatter diagram showing carrier phase synchronization.

Symbol timing recovery can also be observed by looking at the scatter diagram. Figure 6.25 shows how the constalletion diagram start with 16 constellation points, four of which lie at the origin, and then converge to only four.

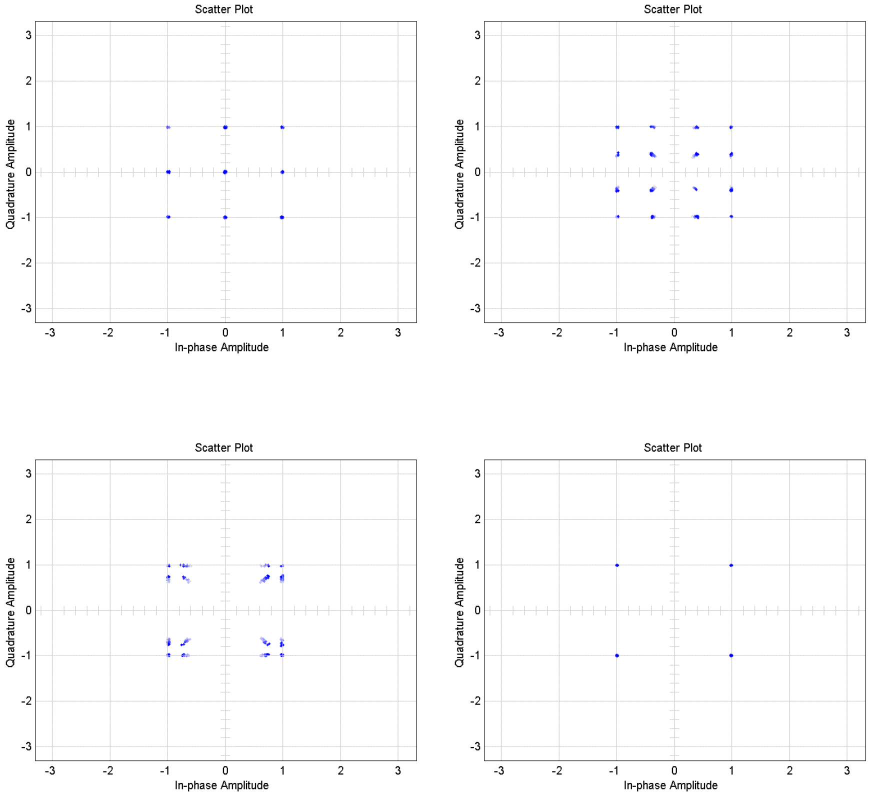


Figure 6.25: Scatter diagram showing carrier phase synchronization.

Symbol error rate (SER) simulations are done to evaluate the performance of the simulation model. First an open loop simulation is done with no differential encoding is show in figure 6.26. The degradation in performance when compared to the theoretical limit is caused by the approximation of the matched filter. The rest of the traces are the closed loop simulations of the three different phase error detectors with differential encoding. There does not seem to be a noticeable difference in the performance of the different PED's.

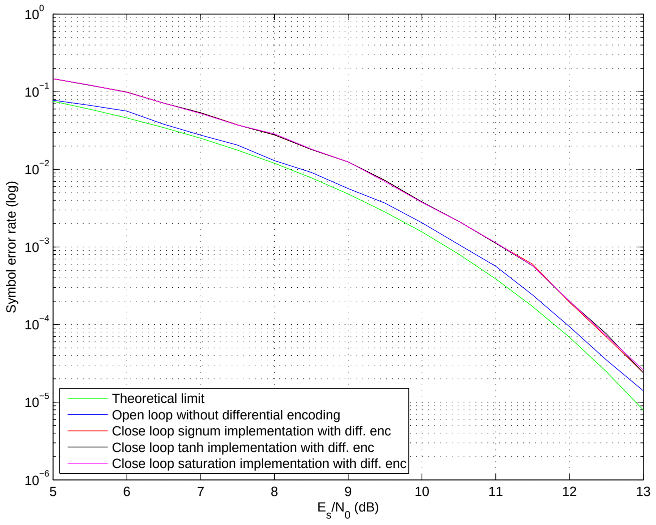


Figure 6.26: Demodulator symbol error rate simulations.

6.4.1 Conclusion

The QPSK demodulator model has been successfully simulated showing how the carrier and symbol recovery loops lock simultaneously and the lock detectors function. The timing recovery simulations were not included in this chapter as they are identical to the results of section 4.1. The carrier recovery loop were simulated and the step responses are close to the designed loop bandwidths. The step responses of the internal and external recovery show higher than expected overshoot which means the damping factor is too low. Additional investigation is needed to determine the cause but one probable cause could be due to the linearized model which is only an approximation. The overshoot is typically not as important as the loop bandwidth. The symbol error rate simulations correlate well with what the theory predict. The biggest degradation of the symbol error rate is due to the matched filter used and the differential encoding.

Chapter 7

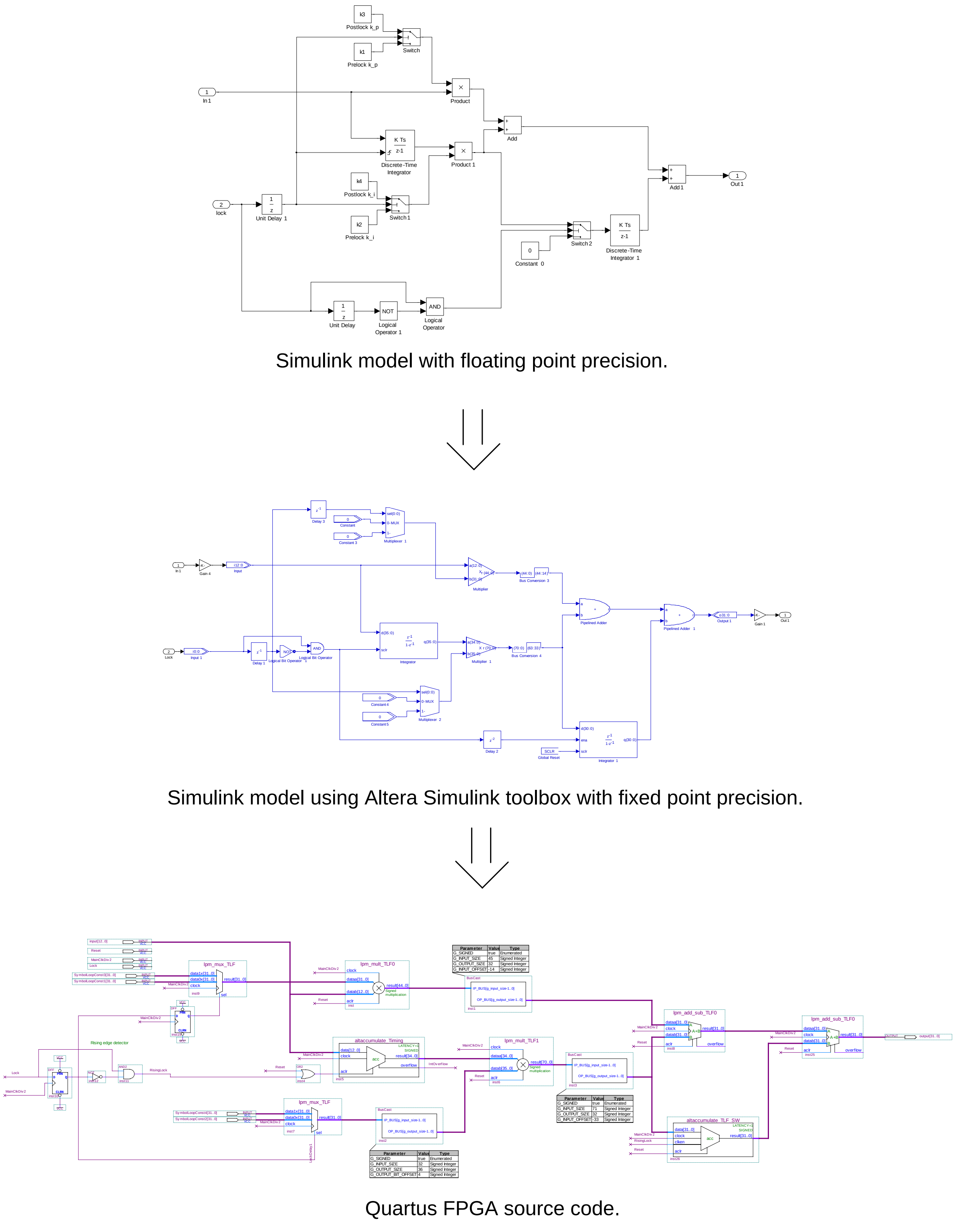
FPGA firmware development

7.1 Demodulation overview

The FPGA software was programmed with Quartus 9 from Altera. The Matlab Simulink simulations were used to develop the algorithms necessary to extract the data from a QPSK signal. This simulation environment was very convenient in the sense that floating point arithmetic and variables could be used. The general process that followed was to convert the Simulink blocks incrementally to fixed point Simulink block from the Altera toolbox which is cycle accurate representation of what the standard Altera firmware blocks would produce. Finally the FPGA code was written based on the Simulink model with the Altera blocks. Figure 7.1 shows how the timing loop filter progressed from the floating point Simulink model to the final FPGA blocks.

The QPSK modulator logic is clocked by a fixed 100 MHz crystal oscillator. The demodulator logic is clock by the symbol recovery DDS clock. A Nios II soft-core processor is used to calculate and initialize the feedback coefficients of the feedback loops. This created a versatile platform where loop parameters could specified in C code and the processor would calculate the coefficients without needing to recompile the FPGA firmware which takes 10 minutes to compile on a 3.2 GHz Core 2 Duo Intel processor with 4 gigabytes of memory.

The numerically controlled oscillator was implemented with an Altera IP core which can be used in evaluation mode. The only limitation is that a flash file cannot be generated which means the demodulator must be programmed with Quartus each time it is turned on. The FPGA firmware is included in PDF format on the enclosed CD.



Simulink model with floating point precision.

Simulink model using Altera Simulink toolbox with fixed point precision.

Quartus FPGA source code.

Figure 7.1: QPSK demodulator hardware.

Chapter 8

Measurements

This chapter documents the measurements done of the QPSK modulator and demodulator blocks. First the modulator measurements are given followed by the demodulator ones.

8.1 QPSK modulator measurements

The DAC's on the FPGA development board generates the I and Q baseband signals for the QPSK modulator. The modulator baseband waveforms are shown in figures 8.1 and 8.2.

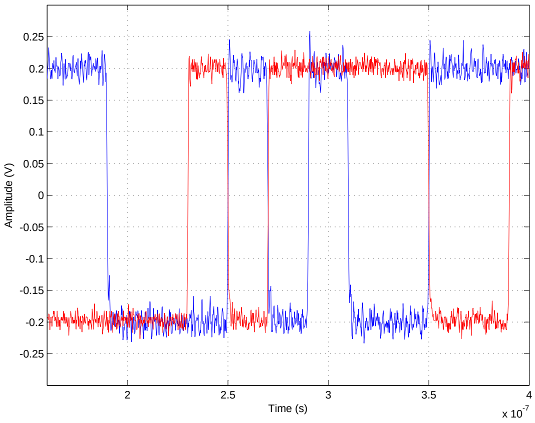


Figure 8.1: I and Q DAC random data waveforms.

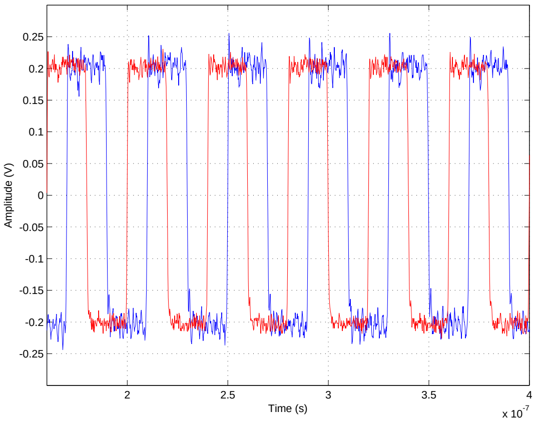


Figure 8.2: I and Q phase shifted waveforms for single sideband generation.

Since the baseband data signals have rectangular pulse shapes the passband QPSK frequency spectrum has the distinct $sin(x)/x$ as shown in figure 8.3. The performance of an I/Q modulator is typically measured with its single sideband rejection ratio. Figure 8.4 shows the SSB measurement with using the baseband data signals from figure 8.2. The baseband SSB modulating signals are 89.4° out of phase. The single sideband rejection ratio is measured as 32 dB and the carrier rejection 38 dB relative to the lower sideband. The 720 MHz synthesizer that generates carrier for the QPSK modulator has several harmonics and spurious components which are

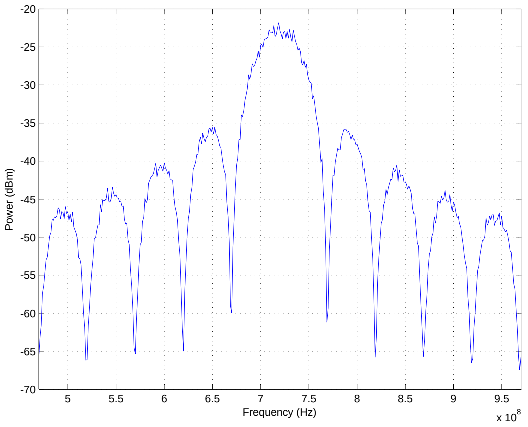


Figure 8.3: Modulator QPSK frequency spectrum.

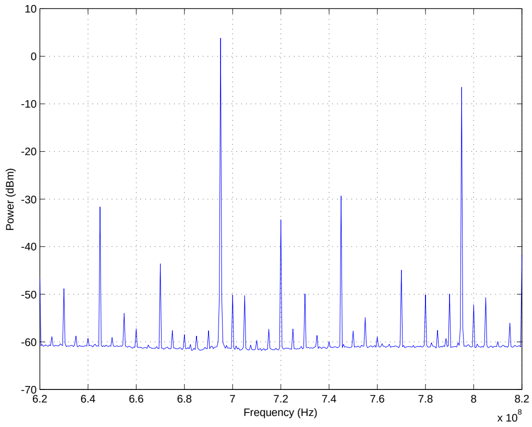


Figure 8.4: Modulator single sideband frequency spectrum.

shown in figure 8.5. The harmonics especially have a big influence on the SSB performance of the modulator.

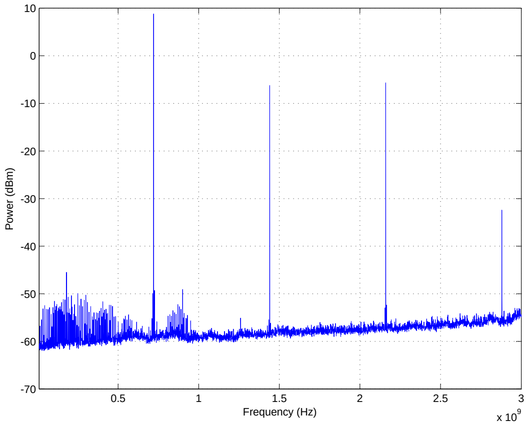


Figure 8.5: 720 MHz synthesizer frequency spectrum unfiltered.

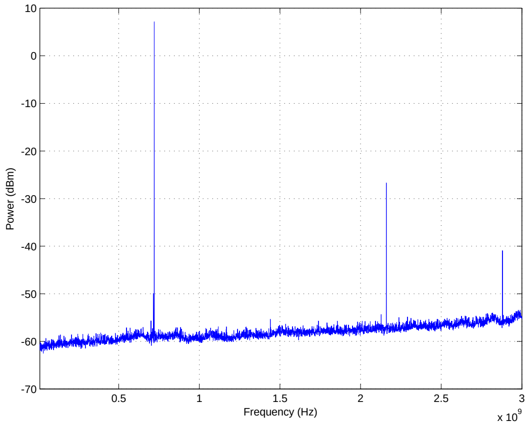


Figure 8.6: 720 MHz synthesizer frequency spectrum filtered.

The synthesizer output filter suppresses most of the spurious signals leaving only some harmonics which are low enough as to not noticeably impact the SSB measurements. Figure 8.7 shows the phase noise of the synthesizer measured on a spectrum analyzer with a resolution bandwidth of 10 Hz.

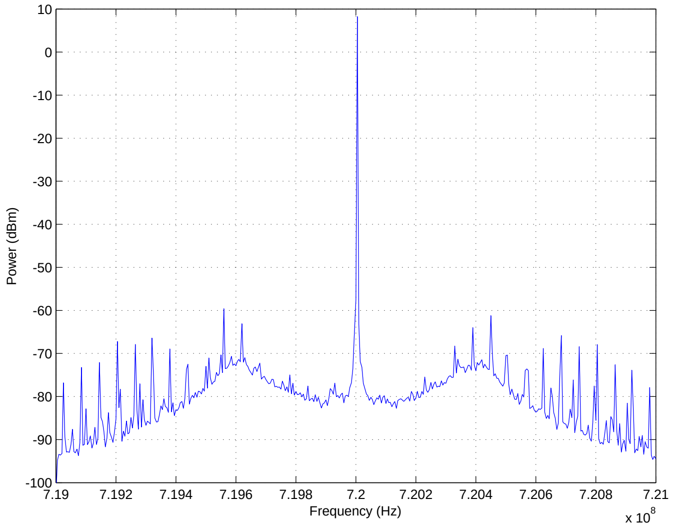


Figure 8.7: 720 MHz synthesizer phase noise.

8.2 QPSK demodulator measurements

The first couple of measurements of the demodulator measure the performance of the analog RF components followed by overall system measurements. The same synthesizer as the modulator is used for the demodulator. The output frequency of the synthesizer is 1240 MHz and is mixed with the 200 MHz of the carrier recovery DDS. The synthesizer output frequency spectrum is shown in figure 8.8. The harmonics of the 1240 MHz carrier are left unfiltered as they are spaced far enough to not create mixing products close to the wanted 1440 MHz.

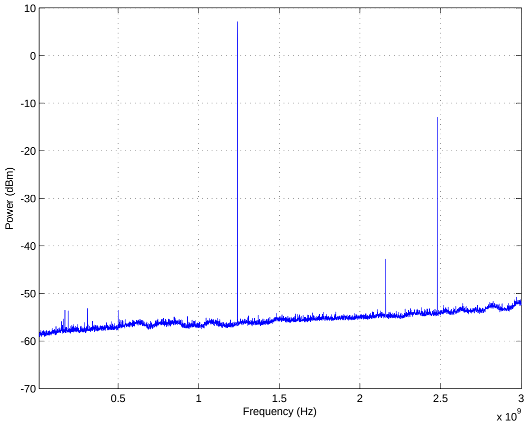


Figure 8.8: 1240 MHz synthesizer frequency spectrum unfiltered.

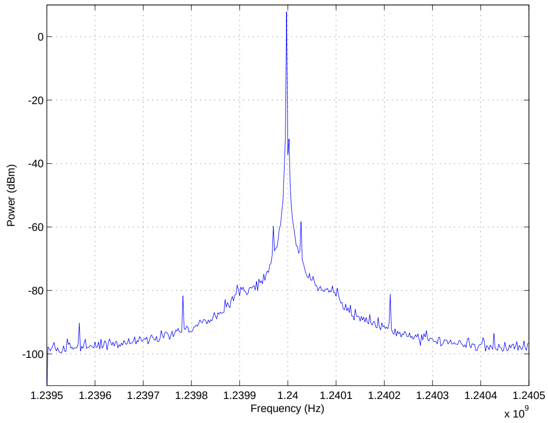


Figure 8.9: 1240 MHz synthesizer phase noise.

The phase noise of the demodulator synthesizer is -97 dBc which is fairly good for such a cheap single IC solution. The carrier recovery DDS generates a variable carrier for coherent down conversion. The DDS output frequency spectrum is shown in figure 8.10.

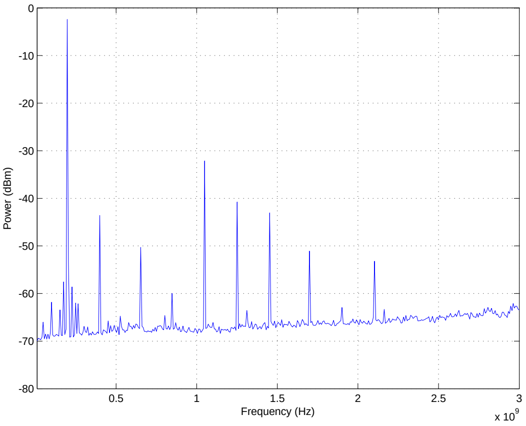


Figure 8.10: Carrier recovery DDS frequency spectrum.

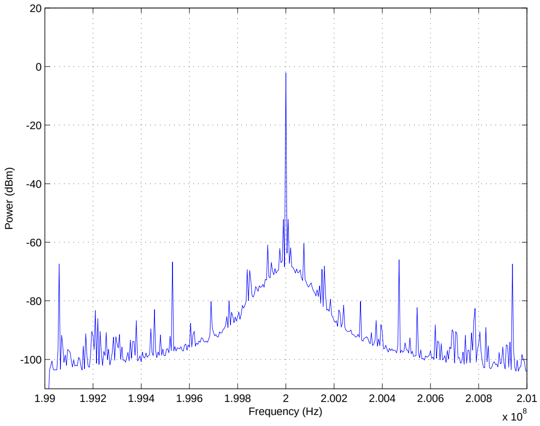


Figure 8.11: Carrier recovery DDS phase noise.

The DDS board is socketed onto the FPGA development board which generates a lot of noise with its DC-DC converters in the power supply lines and the ground. The 470 kHz spurs from the DC-DC converters are clearly visible in the phase noise measurement in figure 8.11. The demodulator synthesizer and carrier recovery DDS go to a mixer which generates the 1.44 GHz carrier for the IQ down converter. The mixer output is filtered by a SAW filter which suppresses most of the unwanted mixing products. See figure 8.12.

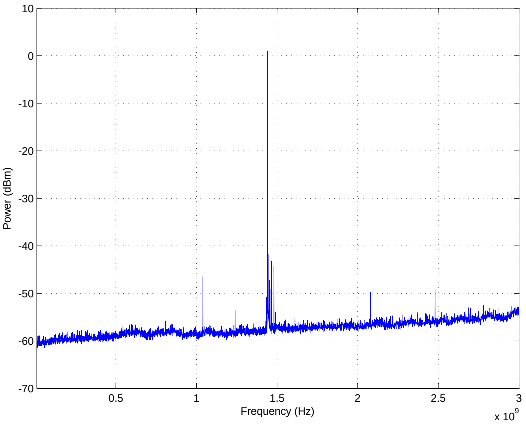


Figure 8.12: Carrier recovery mixer output frequency spectrum.

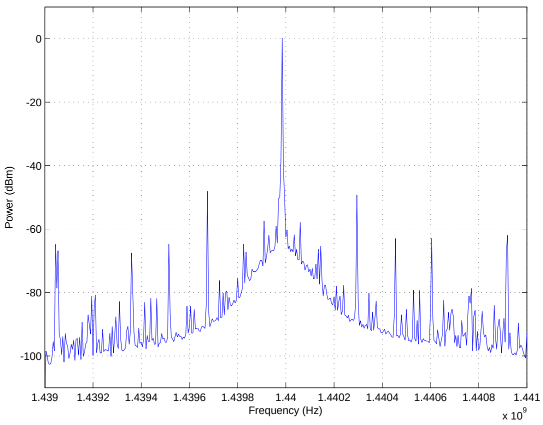


Figure 8.13: Carrier recovery mixer output phase noise.

The mixer output phase noise is dominated by the phase noise from the carrier recovery DDS.

The symbol recovery DDS generates the clock for the analog to digital converters to implement synchronous sampling. Thus any jitter on the DDS clock translates to jitter of the samples taken by ADC's. The output of the DDS goes through a filter and a clock distribution IC which creates the rectangular clock waveform as shown in figure 8.14.

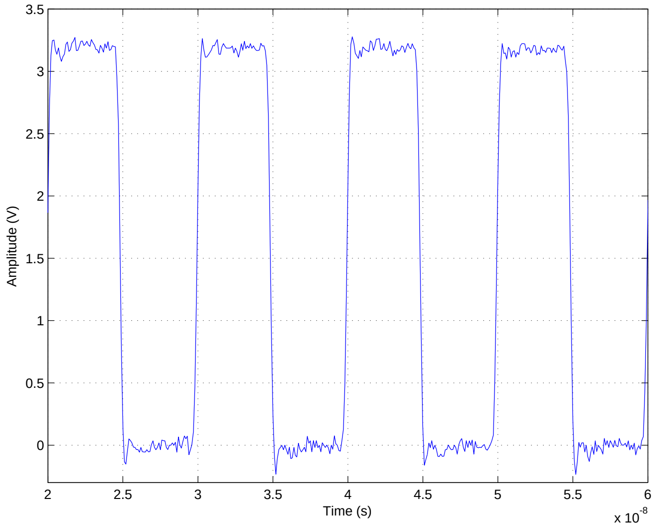


Figure 8.14: DDS generated clock for ADC's.

The I/Q demodulator generates the baseband I and Q channels for the ADC's. First alternating ones and zeros are generated by the modulator and then down converted. The down converted waveforms are shown in figure 8.15. The amplitude is only $80mV_{Peak}$ which results in some measurement noise. There was a problem with the on-board ADC's of the FPGA development board where the positive quantization intervals are bigger than the negative intervals. The problem only seems to manifest itself with large input signals so to avoid the problem only a small range of the ADC input is used. It will increase the quantization noise but it was good enough to test the system. It might be caused by an impedance imbalance at the input of the ADC's due to the removal of the transformers at the input. The transformers lower cut off frequency of 1 MHz created distortion that looks similar to intersymbol interference.

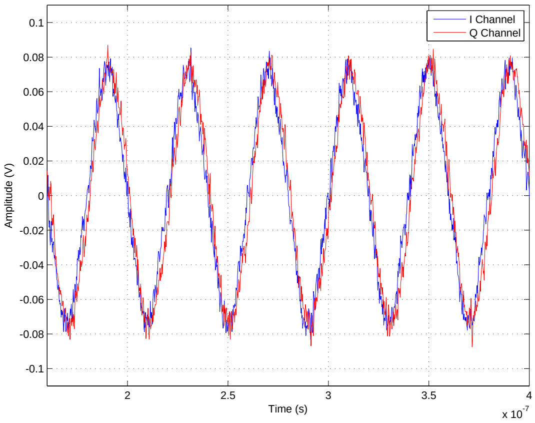


Figure 8.15: I/Q demodulator alternating baseband signals.

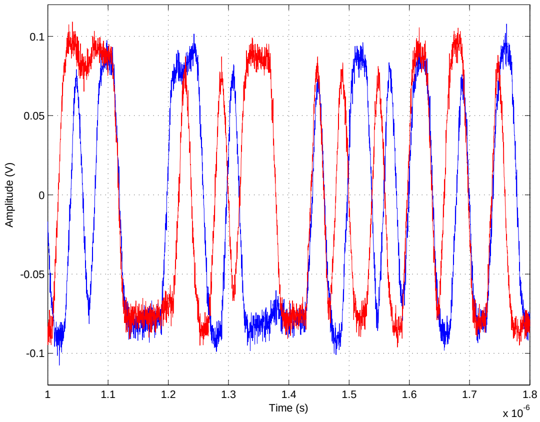


Figure 8.16: IQ demodulator baseband signals with random data.

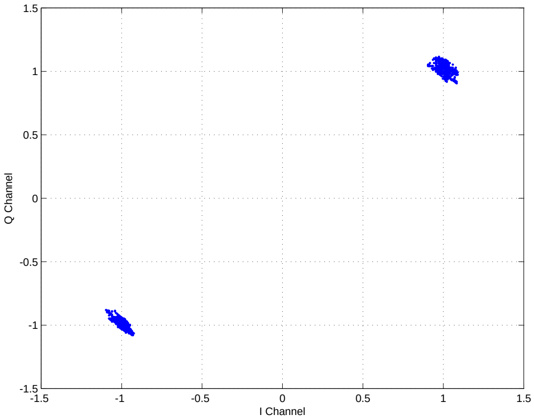


Figure 8.17: Scatter diagram for alternating data.

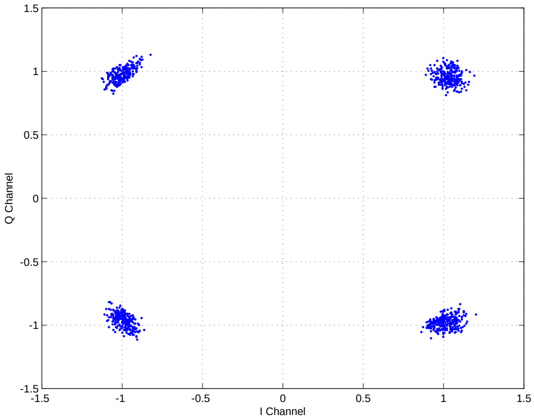


Figure 8.18: Scatter diagram for random data.

Random data was generated in figure 8.18. There is a little cloud around each constellation point which shows the variation of the symbol peaks. Since noise is not injected into the IF signal the problem lies with the implementation of the hardware. There are several implementation problems that cause the amplitude variation which is listed below:

1. AC coupling of the baseband signals - It usually difficult to DC couple the baseband I and Q channels to I/Q modulators and demodulators. Wideband operational amplifiers with gigahertz bandwidths can usually not be DC coupled. The result is if several of the same symbols are received, the amplitude starts to droop reducing the signal amplitude.

2. The matched filter frequency response - The implementation of the matched filter with discrete components has the problem that components come in only standard values but more problematic is the high frequency performance of the inductors that are used. The

inductance is usually less at higher frequencies which lowers the bandwidth of the filter and causes the peak values of two alternating ones or zeros to be less than when two of the same consecutive bits are received.

3. The input match of I/Q demodulator - The I/Q demodulator has two inductors at the RF input for matching. It is very important to use high quality inductors for the match but even if perfect inductors were used the matching network itself will have a limited bandwidth. This becomes a problem for broadband signals where the ratio of the bandwidth to center frequency is more than 15%.

4. The phase noise of the modulator and demodulator LO - Any jitter in the LO would cause slight rotations of the constellation diagram.

When looking at the scatter diagrams one can see that the constellation diagram has some phase noise by the rotational jitter. Next the step response of the timing loop is measured. The demodulator is first allowed to lock. A trigger is setup inside the FPGA firmware to download the timing loop filter output through the SignalTap II Logic Analyzer in Quartus. A $\frac{\pi}{2}$ phase shift is made by delaying the modulator data by half a bit period. The output is shown in figure 8.19.

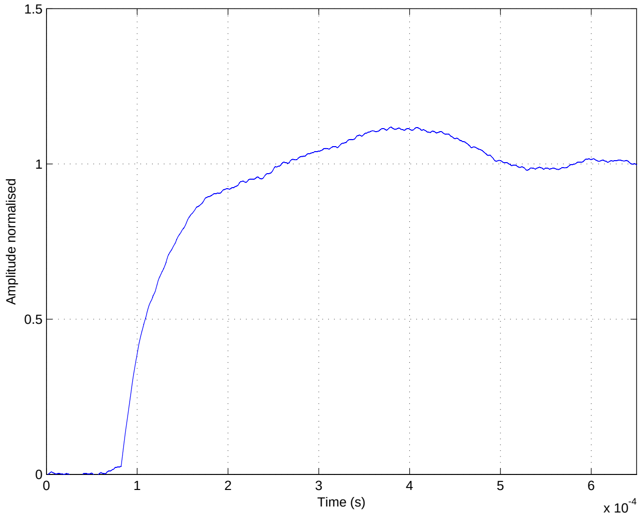


Figure 8.19: Timing recovery loop phase step response measurement.

The phase step causes the timing error detector to operate in its non linear region which makes it a bit difficult to accurately determine the loop bandwidth. Using the peak value and the time at the peak the effective loop bandwidth is calculated as 6.4 kHz. Comparing this result to results of simulations done with similar input phase steps, the loop bandwidth should be about 8.5 kHz for a designed loop bandwidth of 25 kHz. Multiplying the the 6.4 kHz by the ratio of the designed bandwidth to the simulated bandwidth one get 18.8 kHz. Next the internal carrier step response is measured. The phase step is generated by switching the lookup table of modulator constellation diagram to a slightly rotated constellation diagram.

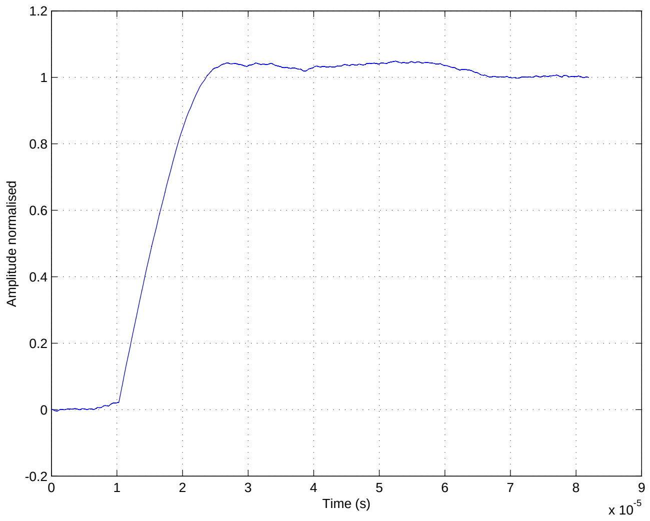


Figure 8.20: Carrier recovery loop phase step response measurement.

The carrier phase step response has 'n large flat overshoot which is open to interpretation where the peak is since this loop response is clearly not a linear second order system. Using the first overshoot the bandwidth is calculated as 86 kHz which fairly close to the designed 100 kHz. The step responses of the carrier symbol recovery loops were made by minimizing the loop bandwidths of the other feedback systems so as to minimize their influence on the step response.

8.2.1 Conclusion

The measurements done show the QPSK modulator generates QPSK and has a SSB rejection ratio of 32 dB which is acceptable. The demodulator recovers the baseband I and Q signals but have some measurement noise due to their small amplitudes. The loop bandwidth of the symbol recovery loop was measured as 18.8 kHz. The carrier recovery loop bandwidth is measured as 86 kHz for a designed bandwidth of 100 kHz. Implementation problems pertaining to the analog hardware was discussed and recommendations will be made in the next chapter. Bit error rate measurements were not made as they are primarily dependent on the implementation of the RF components and was not the main focus of this thesis.

Chapter 9

Conclusion and Recommendations

This thesis discussed the design of a high speed architecture for a QPSK demodulator. The main focus was on developing synchronization techniques for carrier and symbol recovery. A QPSK modulator was also built to generate a QPSK signal for the demodulator. This project required multi disciplinary knowledge from analog RF design, digital signal processing, FPGA programming and control systems with each their own challenges. The RF design and FPGA programming proved to be particularly problematic. A Simulink model for the modulator and demodulator to simulate the demodulation process and verified the recovery algorithms before they were implemented in firmware.

The I/Q demodulator board which frequency translates the QPSK signal to baseband was a critical board in the system. The RF input of the I/Q demodulator IC requires two inductors for matching and reduced the eye diagram of the I/Q demodulator outputs when lower quality inductors were used which cause a slight input mismatch. RF matching networks also have limited bandwidths and when the required bandwidth to center frequency ratio of the IF is higher than 15% or 20% other techniques need to be considered to match subsystems. Resistive matching is very broadband but sacrifices gain which is typically not a problem at IF stages. The matched filters at the I/Q demodulator baseband outputs are probably the most important filters in the demodulator to influence the symbol error rate. It is very important to use high frequency inductors for the matched filter such as the B82496C series from EPCOS. When lower quality inductors are used, the frequency response of the filter is lower than what it should be which causes a form of intersymbol interference.

Synchronous baseband sampling of the matched filter outputs was implemented for symbol recovery by using a 1 GSPS DDS to generate a variable clock for the ADC's. The programming of this DDS presented a couple of problems. The 16 bit parallel programming bus meant that only 16 bits of the 32 bit tuning word could be programmed. This created problems when the integrators went beyond the range of the programming bus during symbol timing acquisition and could not recover from this situation. Range limiting could solve this problem but it still limits the lock range of the symbol recovery loop. The timing error detector (TED) in the symbol recovery loop requires two samples per symbol and generates self noise when there are no bit transitions. The zero crossing TED had much less self noise than the early late TED or the Gardner TED. The self noise is compounded when the frequency update rate of the DDS is lower than the TED clock rate. The lower update rate is due to the programming delay of the DDS. A solution was presented where the TED output could be averaged while the DDS is being programmed instead of discarding the samples. This is however not simple to implement because the clock domain of the DDS is different to that of the FPGA. In light of the above mentioned problems it is recommended that a numerically controlled oscillator be implemented inside the FPGA and fed to a high speed DAC to generate the clock. This ensures one clock domain for all the recovery logic and does not have the 16 bit bus size limitation. The loop bandwidth of the symbol recovery loop was measured as 18.8 kHz with an original designed bandwidth of 25 kHz. Step response has non-linear elements which makes the measurement a bit more difficult.

A theoretical derivation of a maximum likelihood phase estimator for a QPSK signal was done in addendum B. The resulting equation shows a configuration similar to a typical Costas loop except the signum functions are in actual fact tanh functions. In the control loop there is also gain blocks right before the tanh funtions which are proportional to the signal to noise ratio. Since the tanh is a non-linear function the gain has a significant effect on the result of the tanh function. The tanh function could be better approximated by a saturation function but due to the requirement of the known noise level for the gain block it was decided to implement the tried and tested signum function which is not affected by the noise scaling. The scaling of the loop gain by the signal to noise ratio would also change the loop bandwidth which will need compensation. The internal carrier recovery loop bandwidth was designed to be 100 kHz and the measurement showed it to be 95 kHz. The external carrier recovery loop was designed to have a 100 kHz pre-locking bandwidth and was measured as 86 kHz.

The digital AGC which was designed and used only the pulse peak samples to calculate the baseband power but this configuration produced stability problems when the timing recovery loop is not locked. The limit integrator range is also partly responsible for this problem. The power should rather be calculated from the peak and zero crossing samples but was not further investigated.

Simulation showed that the matched filter is the most critical block to influence the symbol error rate (SER). In the simulations the 9th order matched filter degraded the $\frac{E_s}{N_0}$ by 0.2 dB. The carrier and symbol recovery degrades the performance of the demodulator by about 0.02dB when comparing the (SER) to ideal open loop simulations. Differential encoding was also implemented to remove the $\frac{\pi}{2}$ phase ambiguity but doubles the symbol error rate. SER measurements were not made as that was beyond the scope of the project and is primarily dependent on the implementation of the RF subsystems and not the carrier and symbol recovery algorithms.

Appendices

Appendix A

Phase lock loop theory

A.1 Continues Time Phase Locked Loops

Phase locked loops were introduced as early as the 1930's. One of the primary uses for PLL's is in coherent carrier recovery. Phase locked loop theory is very well documented, but for completeness, the basic math will be discussed. The phase locked loop could be shown as a maximum likelihood phase estimator. This is shown in [4]. The phase detector measures the phase difference between the input signal and the VCO and generates an error signal which typically has $\sin(\theta - \hat{\theta})$ coefficient. The PLL in figure A.1 could be restructured into a phase equivalent model shown in figure A.2. The sine function is non-linear but could be linearized when the phase

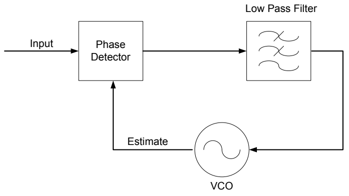


Figure A.1: Basic phase lock loop topology.

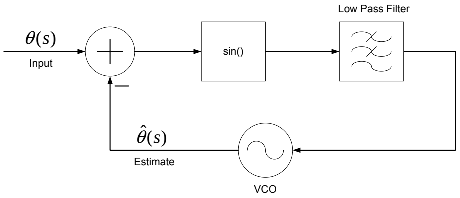


Figure A.2: Basic phase lock loop topology in S domain.

difference is small.

$$\sin(\theta - \hat{\theta}) \approx \theta - \hat{\theta} \quad \text{when} \quad \theta - \hat{\theta} \ll 1 \tag{A.1.1}$$

The voltage controlled oscillator could be approximated as free integrator with a gain. Since frequency is the derivative of phase, we have to integrate the frequency input to get the phase equivalent model. The constant k_0, is the VCO sensitivity.

$$\hat{\theta}(s) = \mathscr{L}\left\{k_0 \int_0^\infty v_c(t) \cdot dt\right\} = \frac{k_0}{s} \cdot v_c(s) \tag{A.1.2}$$

The derivation of the PLL in [4], shows an integrator as the loop filter. This is optimum when there is no frequency offset, but to improve the loop response for frequency offsets, it is preferable to use a first or second order loop filter. A typical filter used is a proportional plus integral filter, shown in [8].

$$F(s) = k1 + \frac{k2}{s} \tag{A.1.3}$$

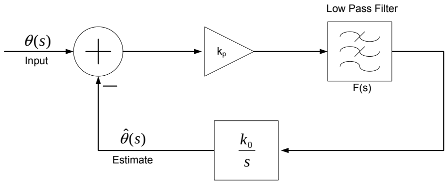


Figure A.3: Linearised phase lock loop.

Using figure A.3, the system transfer function can now be determine.

$$\begin{aligned} H_a(s) = \frac{\hat{\theta}(s)}{\theta(s)} &= \frac{k_0 k_d F(s)}{s + k_0 k_d F(s)} && \text{(A.1.4)} \\ &= \frac{k_p k_0 k_1 s + k_p k_0 k_2}{s^2 + k_p k_0 k_1 s + k_p k_0 k_2} && \text{(A.1.5)} \\ &= \frac{2\zeta\omega_n s + \omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} && \text{(A.1.6)} \end{aligned}$$

From equation A.1.5 and equation A.1.6 the constants are related as follows:

$$\begin{aligned} k_p k_0 k_1 &= 2\zeta\omega_n && \text{(A.1.7)} \\ k_p k_0 k_2 &= \omega_n^2 && \text{(A.1.8)} \end{aligned}$$

The PLL loop dynamics is usually specified by ζ and ω_n, which is the damping factor and natural resonant frequency respectively. Another common parameter that is specified, is the equivalent noise bandwidth. This parameter is defined as the bandwidth of an ideal low-pass filter whose output power is the same as the linear system when excited with white noise.

$$B_L = \int_0^\infty \mid H(j2\pi f) \mid^2 df \tag{A.1.9}$$

$$= \frac{1}{2}\omega_n(\zeta + \frac{1}{4\zeta}) \tag{A.1.10}$$

The noise bandwidth parameter could now be related to the loop constants.

$$k_p k_0 k_1 = \frac{4\zeta B_n}{\zeta + \frac{1}{4\zeta}} \tag{A.1.11}$$

$$k_p k_0 k_2 = \frac{4B_n^2}{(\zeta + \frac{1}{4\zeta})^2} \tag{A.1.12}$$

Another characteristic of a PLL is its acquisition time. The are two processes that determine the acquisition time. First, the time it takes to obtain frequency lock and after that phase lock. For a second order system, the acquisition times are well approximated by the following:

$$T_{FL} \approx 4\frac{(\Delta f)^2}{B_n^3} \tag{A.1.13}$$

$$T_{PL} \approx \frac{1.3}{B_n} \tag{A.1.14}$$

It can be seen that the larger the loop bandwidth, the faster the acquisition of the PLL. The frequency offset between the reference signal and the VCO (Δf), cannot be arbitrarily large. There is a upper limit which determines the pull-in range.

$$\Delta f \leq (2\pi\sqrt{2}\zeta)B_n \approx 6B_n \tag{A.1.15}$$

Finally, the last characteristic that will be discussed of the continues time PLL, is the tracking performance. The Cramer-Rao inequality can be used to show that there is a lower bound for the variance of of the phase estimate $\hat{\theta}_{ML}$. This derivation is also shown in [4].

$$\text{var}\,\{\hat{\theta}_{ML}\} \geq \frac{N_0 B_L}{P_s} \tag{A.1.16}$$

The noise power spectral density was assumed as $\frac{N_0}{2}$ W/Hz. From equation A.1.16, it is clear that the variance of the phase estimate is inversely proportional to the loop signal to noise ratio. It should be noticed that the variance is decreased when the bandwidth is reduced but is done at the expense of the acquisition time. This is a very intuitive result.

Appendix B

QPSK Phase Estimation

The following derivation assumes rectangular pulse shaping of the data in the continuous time domain. A QPSK signal can be mathematically expressed as follows:

$$y(t) = \sqrt{2}A \cdot \cos(\omega_c t + \frac{\pi}{2} \cdot i + \frac{\pi}{4} + \theta) + n(t) \quad \text{with } i = 0,1,2,3 \tag{B.0.1}$$

Equation B.0.1 can be expanded into two orthogonal basis functions.

$$\begin{aligned} y(t) &= \sqrt{2}A \cdot \cos(\omega_c t) \cdot [\cos(\frac{\pi}{2} \cdot i + \frac{\pi}{4}) \cdot \cos(\theta) - \sin(\frac{\pi}{2} \cdot i + \frac{\pi}{4}) \cdot \sin(\theta)] \\ & +\sqrt{2}A \cdot \sin(\omega_c t) \cdot [-\sin(\frac{\pi}{2} \cdot i + \frac{\pi}{4}) \cdot \cos(\theta) - \cos(\frac{\pi}{2} \cdot i + \frac{\pi}{4}) \cdot \sin(\theta)] \\ & +n(t) \end{aligned} \tag{B.0.2}$$

The matched filter output in the receiver can expressed as the correlation of the QPSK signal with the orthonormal basis functions. Assume for simplicity $T_s = \frac{N}{f_c}$ where N is an integer.

$$Z_l = \int_0^{T_s} y(t) \cdot \phi_l(t) dt \tag{B.0.3}$$

$$\begin{aligned} Z_1 &= \int_0^{T_s} y(t) \cdot \sqrt{\frac{2}{T_s}} \cos(\omega_c t) dt \\ &= \frac{1}{2}\sqrt{\frac{2}{T_s}} A \cdot T_s \left[\cos(\frac{\pi}{2} \cdot i + \frac{\pi}{4}) \cos(\theta) - \sin(\frac{\pi}{2} \cdot i + \frac{\pi}{4}) \sin(\theta)\right] + N_1 \\ &= \sqrt{E_s} \cdot \left[\cos(\frac{\pi}{2} \cdot i + \frac{\pi}{4}) \cos(\theta) - \sin(\frac{\pi}{2} \cdot i + \frac{\pi}{4}) \sin(\theta)\right] + N_1 \end{aligned} \tag{B.0.4}$$

$$\begin{aligned} Z_2 &= \int_0^{T_s} y(t) \cdot \sqrt{\frac{2}{T_s}} \sin(\omega_c t) dt \\ &= -\sqrt{E_s} \cdot [\sin(\frac{\pi}{2} \cdot i + \frac{\pi}{4}) \cos(\theta) + \cos(\frac{\pi}{2} \cdot i + \frac{\pi}{4}) \sin(\theta)] + N_2) \end{aligned} \tag{B.0.5}$$

To develop a maximum likelihood estimator, the likelihood function of the estimated parameter is needed. A likelihood function is similar to a conditional PDF. It describes the likelihood of a parameter given an observation. When the noise at the receiver is assumed to be additive white Gaussian noise with amplitude $\frac{1}{2}N_0$, an error vector of the QPSK signal can be used to express a likelihood function. Since the basis functions with

$$f_{\mathbf{Z}/\theta,H_i}(z_1,z_2/\theta,H_i) = \frac{1}{\sqrt{\pi N_0}} \exp^{-\frac{1}{N_0}[(z_1-\bar{Z}_1)^2+(z_2-\bar{Z}_2)^2]} \tag{B.0.6}$$

Substituting the expected values of Z_1 and Z_2 results in:

$$\begin{aligned} f_{\mathbf{Z}/\theta,H_i}(z_1,z_2/\theta,H_i) = \frac{1}{\sqrt{\pi N_0}} & \exp^{-\frac{1}{N_0}[(z_1-\sqrt{E_s}(\cos(\frac{\pi}{2}\cdot i+\frac{\pi}{4})\cos(\theta)-\sin(\frac{\pi}{2}\cdot i+\frac{\pi}{4})\sin(\theta)))]^2} \\ & \cdot \exp^{-\frac{1}{N_0}[(z_2+\sqrt{E_s}(\sin(\frac{\pi}{2}\cdot i+\frac{\pi}{4})\cos(\theta)+\cos(\frac{\pi}{2}\cdot i+\frac{\pi}{4})\sin(\theta)))^2]} \end{aligned} \tag{B.0.7}$$

Expanding equation B.0.7 results in a very big expression so lets consider equation B.0.6 and brake it up into different parts. First $[z_1 - \bar{Z}_1]^2$ is expanded.

$$\begin{aligned} [z_1-\bar{Z}_1]^2 &= z_1^2 + 2z_1\sqrt{E_s}[-\cos(\frac{\pi}{2}\cdot i+\frac{\pi}{4})\cos(\theta) + \sin(\frac{\pi}{2}\cdot i+\frac{\pi}{4})\sin(\theta)] \\ &+ E_s\cos^2(\theta)\cos^2(\frac{\pi}{2}\cdot i+\frac{\pi}{4}) - 2E_s\cos(\theta)\sin(\theta)\cos(\frac{\pi}{2}\cdot i+\frac{\pi}{4})\sin(\frac{\pi}{2}\cdot i+\frac{\pi}{4}) \\ &+ E_s\sin^2(\frac{\pi}{2}\cdot i+\frac{\pi}{4})\sin^2(\theta) \end{aligned} \tag{B.0.8}$$

Now $[z_2 - \bar{Z}_2]^2$ is expanded.

$$\begin{aligned} [z_2-\bar{Z}_2]^2 &= z_2^2 + 2z_2\sqrt{E_s}[\sin(\frac{\pi}{2}\cdot i+\frac{\pi}{4})\cos(\theta) + \cos(\frac{\pi}{2}\cdot i+\frac{\pi}{4})\sin(\theta)] \\ &+ E_s\cos^2(\theta)\sin^2(\frac{\pi}{2}\cdot i+\frac{\pi}{4}) + 2E_s\cos(\theta)\sin(\theta)\cos(\frac{\pi}{2}\cdot i+\frac{\pi}{4})\sin(\frac{\pi}{2}\cdot i+\frac{\pi}{4}) \\ &+ E_s\cos^2(\frac{\pi}{2}\cdot i+\frac{\pi}{4})\sin^2(\theta) \end{aligned} \tag{B.0.9}$$

Summing equations B.0.8 and B.0.9 and factorising:

$$\begin{aligned} [z_1-\bar{Z}_1]^2 + [z_2-\bar{Z}_2]^2 &= 2z_1\sqrt{E_s}[-\cos(\theta)\cos(\frac{\pi}{2}\cdot i+\frac{\pi}{4}) + \sin(\theta)\sin(\frac{\pi}{2}\cdot i+\frac{\pi}{4})] + \\ & 2z_2\sqrt{E_s}[\cos(\theta)\sin(\frac{\pi}{2}\cdot i+\frac{\pi}{4}) + \sin(\theta)\cos(\frac{\pi}{2}\cdot i+\frac{\pi}{4})] + \\ & z_1^2 + z_2^2 + E_s \end{aligned} \tag{B.0.10}$$

The terms z_1, z_2 and E_s are independent of θ and will disappear in the differentiation process later on, so they will be ignored to simplify the derivation. Since we are dealing with a QPSK signal, there are four symbols with distinct phases that need to be considered ($i \in \{0,1,2,3\}$) Looking at equation B.0.10.

$$\begin{aligned}
i = 0 \rightarrow \quad & 2z_1\sqrt{E_s}[-\frac{1}{\sqrt{2}}\cos(\theta) + \frac{1}{\sqrt{2}}\sin(\theta)] + \\
& 2z_2\sqrt{E_s}[\frac{1}{\sqrt{2}}\cos(\theta) + \frac{1}{\sqrt{2}}\sin(\theta)] \qquad \text{(B.0.11)} \\
i = 1 \rightarrow \quad & 2z_1\sqrt{E_s}[\frac{1}{\sqrt{2}}\cos(\theta) + \frac{1}{\sqrt{2}}\sin(\theta)] + \\
& 2z_2\sqrt{E_s}[\frac{1}{\sqrt{2}}\cos(\theta) - \frac{1}{\sqrt{2}}\sin(\theta)] \qquad \text{(B.0.12)} \\
i = 2 \rightarrow \quad & 2z_1\sqrt{E_s}[\frac{1}{\sqrt{2}}\cos(\theta) - \frac{1}{\sqrt{2}}\sin(\theta)] + \\
& 2z_2\sqrt{E_s}[-\frac{1}{\sqrt{2}}\cos(\theta) - \frac{1}{\sqrt{2}}\sin(\theta)] \qquad \text{(B.0.13)} \\
i = 3 \rightarrow \quad & 2z_1\sqrt{E_s}[-\frac{1}{\sqrt{2}}\cos(\theta) - \frac{1}{\sqrt{2}}\sin(\theta)] + \\
& 2z_2\sqrt{E_s}[-\frac{1}{\sqrt{2}}\cos(\theta) + \frac{1}{\sqrt{2}}\sin(\theta)] \qquad \text{(B.0.14)}
\end{aligned}$$

Now that the four cases have been evaluated, we can construct a likelihood function that is independent of the specific symbol received.

$$f_{\mathbf{Z}/\theta}(z_1, z_2/\theta) = \sum_{i=0}^{3} P(H_i) \cdot f_{\mathbf{z}/\theta,H_i}(z_1, z_2/\theta, H_i) \qquad \text{(B.0.15)}$$

The received symbols are assumed to be statistically independent each with an $\frac{1}{4}$ chance of occurring. Substituting equation B.0.7 into equation B.0.15 produces:

$$\begin{aligned}
f_{\mathbf{Z}/\theta}(z_1, z_2/\theta) \quad = \quad & \sum_{i=0}^{3} \frac{P(H_i)}{\sqrt{\pi N_0}} \exp^{-\frac{1}{N_0}[(z_1 - \sqrt{E_s}(\cos(\frac{\pi}{2}\cdot i + \frac{\pi}{4})\cos(\theta) - \sin(\frac{\pi}{2}\cdot i + \frac{\pi}{4})\sin(\theta)))]^2} \\
& \cdot \exp^{-\frac{1}{N_0}[(z_2 - \sqrt{E_s}(\sin(\frac{\pi}{2}\cdot i + \frac{\pi}{4})\cos(\theta) - \cos(\frac{\pi}{2}\cdot i + \frac{\pi}{4})\sin(\theta)))^2]} \qquad \text{(B.0.16)} \\
= \quad & \frac{1}{4\sqrt{\pi N_0}}[\exp^{-\frac{\sqrt{2E_s}}{N_0}[-z_1\cos(\theta) + z_1\sin(\theta) + z_2\cos(\theta) + z_2\sin(\theta)]}] + \\
& \frac{1}{4\sqrt{\pi N_0}}[\exp^{-\frac{\sqrt{2E_s}}{N_0}[z_1\cos(\theta) + z_1\sin(\theta) + z_2\cos(\theta) - z_2\sin(\theta)]}] + \\
& \frac{1}{4\sqrt{\pi N_0}}[\exp^{-\frac{\sqrt{2E_s}}{N_0}[z_1\cos(\theta) - z_1\sin(\theta) - z_2\cos(\theta) - z_2\sin(\theta)]}] + \\
& \frac{1}{4\sqrt{\pi N_0}}[\exp^{-\frac{\sqrt{2E_s}}{N_0}[-z_1\cos(\theta) - z_1\sin(\theta) - z_2\cos(\theta) + z_2\sin(\theta)]}] \qquad \text{(B.0.17)}
\end{aligned}$$

Equation B.0.17 can be factorised into the following form:

$$
\begin{aligned}
f_{\mathbf{Z}/\theta}(z_1, z_2/\theta) \quad &= \quad \frac{1}{4\sqrt{\pi N_0}}[\exp^{-\frac{\sqrt{2E_s}}{N_0}[z_1\sin(\theta)+z_2\cos(\theta)]} + \exp^{-\frac{\sqrt{2E_s}}{N_0}[-z_1\sin(\theta)-z_2\cos(\theta)]}] \times \\
&\quad\quad [\exp^{-\frac{\sqrt{2E_s}}{N_0}[-z_1\cos(\theta)+z_2\sin(\theta)]} + \exp^{-\frac{\sqrt{2E_s}}{N_0}[z_1\cos(\theta)-z_2\sin(\theta)]}] \qquad \text{(B.0.18)}
\end{aligned}
$$

For the maximum likelihood estimator, we are going to take the derivative of the log-likelihood function and set it to zero to find the value of θ that would maximize the log-likelyhood function. The value of θ that maximizes the log-likehood function also maximizes the original likelihood function.

$$\frac{\partial}{\partial\theta}\ln(f_{\mathbf{Z}/\theta}(z_1, z_2/\theta)) = 0 \qquad \text{(B.0.19)}$$

Before the differentiation is done, a substitution must be made. When we take the log of equation B.0.18, the product becomes the sum of two log terms. Inside each log term, the following substitution can be made: $\frac{\exp^x + \exp^{-x}}{2} = \cosh(x)$.

$$
\begin{aligned}
\ln\left(f_{\mathbf{Z}/\theta}(z_1, z_2/\theta)\right) \quad &= \quad \ln\left(\frac{1}{\sqrt{\pi N_0}}\right) + \ln\left[\cosh\left(-\frac{\sqrt{2E_s}}{N_0}[z_1\sin(\theta) + z_2\cos(\theta)]\right)\right] + \\
&\quad\quad \ln\left[\cosh\left(-\frac{\sqrt{2E_s}}{N_0}[z_1\cos(\theta) - z_2\sin(\theta)]\right)\right] \qquad \text{(B.0.20)}
\end{aligned}
$$

Now we are ready to take the partial derivative of equation B.0.20 with respect to θ.

$$
\begin{aligned}
\frac{\partial}{\partial\theta}\ln\left(f_{\mathbf{Z}/\theta}(z_1, z_2/\theta)\right) \quad &= \quad \frac{\sinh\left(C_1\left[z_1\sin(\theta) + z_2\cos(\theta)\right]\right)\cdot C_1\cdot\left[z_1\cos(\theta) - z_2\sin(\theta)\right]}{\cosh\left(C_1[z_1\sin(\theta) + z_2\cos(\theta)]\right)} + \\
&\quad\quad \frac{\sinh\left(C[z_1\cos(\theta) - z_2\sin(\theta)]\right)\cdot C_1\cdot\left[-z_1\sin(\theta) - z_2\cos(\theta)\right]}{\cosh\left(C_1[z_1\cos(\theta) - z_2\sin(\theta)]\right)} \\
&= \quad \tanh\left(C_1\left[z_1\sin(\theta) + z_2\cos(\theta)\right]\right)\cdot C_1\cdot\left[z_1\cos(\theta) - z_2\sin(\theta)\right] - \\
&\quad\quad \tanh\left(C_1\left[z_1\cos(\theta) - z_2\sin(\theta)\right]\right)\cdot C_1\cdot\left[z_1\sin(\theta) + z_2\cos(\theta)\right] \qquad \text{(B.0.21)}
\end{aligned}
$$

C_1 is a constant with value $\frac{\sqrt{2E_s}}{N_0}$. z_1 and z_2 are the outputs of the matched filters. Substituting equation B.0.3 in results in the following:

$$\begin{aligned}
\frac{\partial}{\partial\theta}\ln\left(f_{\mathbf{Z}/\theta}(z_1, z_2/\theta)\right) &= \tanh\left(C_2\left[\int_0^{T_s} y(t)\cdot\cos(\omega_c t)\cdot\sin(\theta)dt + \int_0^{T_s} y(t)\cdot\sin(\omega_c t)\cdot\cos(\theta)dt\right]\right)\cdot \\
&\quad C_2\cdot\left[\int_0^{T_s} y(t)\cdot\cos(\omega_c t)\cdot\cos(\theta)dt - \int_0^{T_s} y(t)\cdot\sin(\omega_c t)\cdot\sin(\theta)dt\right] \\
&\quad -\tanh\left(C_2\left[\int_0^{T_s} y(t)\cdot\cos(\omega_c t)\cdot\cos(\theta)dt - \int_0^{T_s} y(t)\cdot\sin(\omega_c t)\cdot\sin(\theta)dt\right]\right)\cdot \\
&\quad C_2\cdot\left[\int_0^{T_s} y(t)\cdot\cos(\omega_c t)\cdot\sin(\theta)dt + \int_0^{T_s} y(t)\cdot\sin(\omega_c t)\cdot\cos(\theta)dt\right] \\
&= \tanh\left(C_2\int_0^{T_s} y(t)\cdot\sin(\omega_c t+\theta)dt\right)\cdot C_2\cdot\int_0^{T_s} y(t)\cdot\cos(\omega_c t+\theta)dt \\
&\quad -\tanh\left(C_2\int_0^{T_s} y(t)\cdot\cos(\omega_c t+\theta)dt\right)\cdot C_2\cdot\int_0^{T_s} y(t)\cdot\sin(\omega_c t+\theta)dt \quad \text{(B.0.22)}
\end{aligned}$$

The constant C_2 has the value $\frac{2A}{N_0}$. When equation B.0.22 is zero, $\theta = \hat{\theta}_{ML}$

Bibliography

[1] A. N. Mengali, Umberto&D'Andrea, *Synchronization Techniques for Digital Receivers*. Plenum Press, 1997.

[2] F. G. Stremler, *Introduction to communication systems*, 3rd ed. Addison-Wesley publishing company, 1990.

[3] S. Haykin, *An Introduction to Analog and Digital Communications*. Wiley, 1989.

[4] R. E. Zimmer and W. H. Tranter, *Principles of Communications Systems, Modulation and Noise*, 5th ed. John Wiley & Sons, 2002.

[5] F. M. Gardner, "A bpsk/qpsk timing-error detector for sampled receivers," *IEEE Transactions on Communications*, vol. COM-34, no. 5, May 1986.

[6] *A symbol synchronization lock detection and SNR estimator for QPSK, with application to BPSK*, no. 3, IASTED International conference on wireless and optical communication. Banff,Alberta,Canada: Linn,Yair, July 2003.

[7] F. WenJun, J. JingShan, W. ShuanRong, and L. Luan, "Design and performance evaluation of carrier lock detection in digital qpsk receivers," *Communications*, vol. 46, no. 6, June 1998.

[8] F. M. Gardner, *Phaselock Techniques*, 2nd ed. John Wiley & Sons, 1979.